

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE MÁSTER

Ecosistema de Comunicadores Android con Telemandos

Máster Universitario en Ingeniería de Telecomunicación

Autor: Rodado Peribáñez, Adrián
Tutor: Boemo Scalvinoni, Eduardo

FECHA: Junio 2020

ECOSISTEMA DE COMUNICADORES ANDROID CON TELEMANDOS

AUTOR: Adrián Rodado Peribáñez
TUTOR: Eduardo Boemo Scalvinoni

Digital System Laboratory
Dpto. Tecnología Electrónica y de Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2020

Resumen (castellano)

Este Trabajo de Fin de Máster consiste en el diseño, desarrollo e implementación de un ecosistema de dispositivos Android, tales como teléfonos, tabletas y relojes inteligentes, capaces de comunicarse con una regleta de enchufes por telemandos a través de un dispositivo controlador mediante una conexión Wifi, permitiendo un control remoto sobre dispositivos susceptibles a ser alimentados por corriente eléctrica e integrar todo este sistema en la aplicación *Quiero Decirte* [1], un comunicador Android destinado a mejorar la comunicación e interacción de personas con ciertas discapacidades que implican un bajo nivel cognitivo, tales como trastornos del espectro autista o la deficiencia intelectual, mediante la creación de frases útiles con pictogramas y locuciones.

Este proyecto tiene como objetivo facilitar y acomodar tareas de la vida cotidiana tales como encender y apagar cualquier dispositivo electrónico de manera remota y sencilla a través de una regleta de enchufes inteligentes desde una aplicación integrada en el comunicador Android destinado a personas con un nivel cognitivo bajo proporcionándoles autonomía y desarrollando sus capacidades comunicativas. Por ello, la aplicación deberá tener una interfaz intuitiva, con pocos estímulos, sencilla y accesible para que cualquier usuario pueda utilizarla cómodamente independientemente de su nivel cognitivo.

Para desarrollar dicho objetivo será necesario crear una aplicación Android, que contará con una versión individual y una versión integrada en la aplicación *Quiero Decirte*, compatible para teléfonos, tabletas y relojes inteligentes, capaz de comunicarse con un ordenador de placa reducida a través de una conexión Wifi que actuará sobre un circuito de enchufes que conseguirá realizar la función de una regleta inteligente.

La aplicación y la gestión de los enchufes tendrán un alto nivel de personalización mediante una gran lista de ajustes y eventos, con la posibilidad de añadir nombres, pictogramas e imágenes a los enchufes. La actuación sobre los enchufes a través del ordenador de placa reducida será inmediato y seguro, programando un servidor capaz de recibir y enviar información y actuar sobre otros elementos eléctricos.

El diseño ha sido desarrollado con la colaboración conjunta del *Digital System Laboratory* y un equipo de especialistas del departamento de Apoyo y Orientación del colegio Romero Peña de La Solana, en Ciudad Real.



Abstract (English)

This End of Master's work consists of the design, development and implementation of an ecosystem of Android devices such as phones, tablets and smart clocks, capable of communicating with a remote control power strip through a controller device by means of a Wifi connection allowing remote control over devices susceptible to being powered by electrical current and integrating this entire system into the *Quiero Decirte* application [1], an Android communicator aimed to improve communication and interaction for people with certain disabilities involving a low cognitive level, such as autism spectrum disorders or intellectual impairment, by creating useful phrases with pictograms and speech.

This project aims to facilitate and accommodate everyday tasks such as turning on and off any electronic device remotely and easily through a smart socket strip from an application integrated into the Android communicator aimed at people with a low cognitive level providing them with autonomy and developing their communication skills. Therefore, the application should have an intuitive interface, with few stimuli, simple and accessible so that any user can use it comfortably regardless of their cognitive level.

To develop this objective it will be necessary to create an Android application, which will have an individual version and a version integrated into the *Quiero Decirte* application, compatible with smartphones, tablets and watches, capable of communicating with a small board computer through a Wifi connection that will act on a circuit of plugs that will manage to perform the function of a smart strip.

The application and the management of the plugs will have a high level of customization through a large list of settings and events and with the possibility of adding names, pictograms, and images to the plugs. The action on the sockets through the reduced board computer will be immediate and safe, programming a server capable of receiving and sending information and acting on other electrical elements.

The design has been developed with the collaboration of the Digital System Laboratory and a team of specialists from the Support and Orientation Department of the Romero Peña School in La Solana, Ciudad Real.



Palabras clave (castellano)

Android, pictograma, comunicador, discapacidad, autismo, aplicación, regleta inteligente, conexión.

Keywords (English)

Android, pictogram, Communicator, disability, autism, application, smart strip, connection.

Agradecimientos

A mis padres, hermana, familia, amigos y tutor por acompañarme durante estos últimos años, por todo el apoyo brindado y conseguir que todo sea más fácil.

En especial a mi abuela Delfina, por toda la dedicación, cariño y amor que me ha regalado durante tantos años. Todos mis logros siempre irán dedicados a ti.

ÍNDICE DE CONTENIDOS

1 INTRODUCCIÓN.....	5
1.1 MOTIVACIÓN	5
1.2 OBJETIVOS	6
1.3 ORGANIZACIÓN DE LA MEMORIA	7
2 ESTADO DEL ARTE	8
2.1 INTRODUCCIÓN.....	8
2.2 SISTEMAS DISPONIBLES SIMILARES	8
2.2.1 Regletas de enchufes inteligentes.....	8
2.2.2 Aplicaciones Android controladores de regletas de enchufes inteligentes.....	10
3 DISEÑO	13
3.1 OBJETIVOS DE DISEÑO	13
3.2 REQUISITOS DE DISEÑO DE LA APLICACIÓN ANDROID	13
3.2.1 Especificaciones técnicas.....	13
3.2.2 Estructura de la aplicación	16
3.3 REQUISITOS DEL DISEÑO HARDWARE	17
4 DESARROLLO.....	21
4.1 HERRAMIENTAS APLICADAS.....	21
4.2 DESARROLLO DE LA APLICACIÓN ANDROID	21
4.2.1 Versión para móviles y tabletas.....	22
4.2.2 Versión para relojes inteligentes.....	29
4.2.3 Implementación en aplicación Quiero Decirte.....	32
4.3 DESARROLLO DE LA REGLETA DE ENCHUFES INTELIGENTES	33
4.3.1 Elementos de la regleta de enchufes inteligentes.....	33
4.3.2 Diseño del sistema	35
4.4 DESARROLLO DE LA CONEXIÓN ENTRE LA APLICACIÓN ANDROID Y LA REGLETA DE ENCHUFES INTELIGENTES	36
5 INTEGRACIÓN, PRUEBAS Y RESULTADOS.....	41
5.1 INTEGRACIÓN	41
5.1.1 Integración de la aplicación Android.....	41
5.1.2 Integración del circuito de la regleta de enchufes inteligentes.....	42
5.2 PRUEBAS.....	43
5.3 RESULTADOS.....	43
6 CONCLUSIONES Y TRABAJO FUTURO.....	47
6.1 CONCLUSIONES.....	47
6.2 TRABAJO FUTURO	48
REFERENCIAS	49
GLOSARIO	51
ANEXOS	I
A MANUAL DE USUARIO	I
B PANTALLAS DE LA APLICACIÓN	- 1 -

ÍNDICE DE FIGURAS

FIGURA 2-1: REGLETA DE ENCHUFES INTELIGENTES WIFI LOONFREE	9
FIGURA 2-2:REGLETA DE ENCHUFES INTELIGENTES WIFI TAOCOCO	10
FIGURA 2-3: LOGOTIPO SMART LIFE – SMART LIVING	11
FIGURA 2-4: LOGOTIPO KASA SMART	12
FIGURA 3-1: TABLETAS Y TELÉFONOS MÁS VENDIDOS SEGÚN SU SISTEMA OPERATIVO	14
FIGURA 4-1 : LOGOTIPO DE LA APLICACIÓN CONTROL REMOTO	22
FIGURA 4-2: PANTALLA DE PRESENTACIÓN DE LA APLICACIÓN	22
FIGURA 4-3: PANTALLA PRINCIPAL DE LA APLICACIÓN Y SU MENÚ LATERAL	23
FIGURA 4-4: FRAGMENTO DE LA INFORMACIÓN DE LA APLICACIÓN	24
FIGURA 4-5: PANTALLA DE INICIO DE SESIÓN DE LA APLICACIÓN	24
FIGURA 4-6: PANTALLA DE AJUSTES DE LA APLICACIÓN	25
FIGURA 4-7: FRAGMENTO PARA LA CONEXIÓN CON LA RASPBERRY PI.....	25
FIGURA 4-8: PANTALLA DE PERSONALIZACIÓN DE ENCHUFES	26
FIGURA 4-9: FRAGMENTO DE SELECCIÓN DEL ENCHUFE A RESTABLECER.....	26
FIGURA 4-10: PANTALLA DE LAS ANALÍTICAS DE LOS ENCHUFES.....	27
FIGURA 4-11: PANTALLA DE GENERACIÓN DE EVENTOS Y ANALÍTICAS	28
FIGURA 4-12: PANTALLA DE PRESENTACIÓN	29
FIGURA 4-13: PANTALLA DE CONTROL DEL ENCHUFE 1	29
FIGURA 4-14: PANTALLA DE ACCESO A LOS AJUSTES DE LA APLICACIÓN	30
FIGURA 4-15: PANTALLA DE AJUSTES DE LA APLICACIÓN.....	30
FIGURA 4-16: PANTALLA DE CONEXIÓN CON LA RASPBERRY PI	30
FIGURA 4-17: PANTALLAS DE PERSONALIZACIÓN DE LOS ENCHUFES	31
FIGURA 4-18: PANTALLA DE GENERACIÓN DE EVENTOS Y NOTIFICACIONES	31
FIGURA 4-19: IMPLEMENTACIÓN DE LA APLICACIÓN EN EL COMUNICADOR QUIERO DECIRTE	33
FIGURA 4-20: MODULO DE RELÉ DE 4 CANALES	34
FIGURA 4-21: BASE DE ENCHUFE MODULAR	35
FIGURA 4-22: ESQUEMÁTICO DEL SISTEMA DE LA REGLETA DE ENCHUFES INTELIGENTES	36
FIGURA 4-23: CÓDIGO PYTHON DEL SERVIDOR EN LA RASPBERRY PI	37
FIGURA 4-24: CÓDIGO DE LA APLICACIÓN ANDROID EN JAVA QUE ESTABLECE LA CONEXIÓN CON LA RASPBERRY PI	39
FIGURA 5-1: BASE DE DATOS DE USUARIOS DE LA HERRAMIENTA FIREBASE	42
FIGURA 5-2: CIRCUITO COMPLETO DE LA REGLETA DE ENCHUFES INTELIGENTES.....	43
FIGURA 5-3: DATOS DE INTERACCIÓN DE LOS USUARIOS DE LA APLICACIÓN EN LA HERRAMIENTA DE FIREBASE.....	44
FIGURA 5-4: RESULTADOS EN LA CONSOLA DE LA RASPBERRY PI Y EN LA INTERFAZ DE LA APLICACIÓN AL ESTABLECER LA CONEXIÓN	45
FIGURA 5-5: RESULTADOS DE LA CONSOLA DE LA RASPBERRY PI AL RECIBIR DATOS DE LA APLICACIÓN ANDROID	45
FIGURA 5-6: RESULTADO DE LA INTERACCIÓN ENTRE LA APLICACIÓN ANDROID Y LA RASPBERRY PI AL INTENTAR ENCENDER UNO DE LOS ENCHUFES DE LA REGLETA INTELIGENTE DE MANERA REMOTA	46
FIGURA B-1: PANTALLAS DEL COMUNICADOR QUIERO DECIRTE – PANTALLA DE AJUSTES CON LA OPCIÓN DE HABILITACIÓN DEL CONTROL REMOTO – PANTALLA PRINCIPAL CON EL ACCESO A LA SECCIÓN DE CONTROL REMOTO	- 1 -
FIGURA B-2: PANTALLA DE PRESENTACIÓN - PANTALLA PRINCIPAL DE CONTROL DE ENCHUFES – PANTALLA PRINCIPAL CON EL MENÚ LATERAL	- 1 -
FIGURA B-3: FRAGMENTO DE LA INFORMACIÓN TÉCNICA DE LA APLICACIÓN – PANTALLA DE INICIO DE SESIÓN O REGISTRO – PANTALLA DE CIERRE DE SESIÓN O VERIFICACIÓN DE EMAIL	- 2 -
FIGURA B-4: PANTALLA DE AJUSTES DEL CONTROL REMOTO - FRAGMENTO DE ESTABLECIMIENTO DE LA CONEXIÓN CON LA RASPBERRY PI – PANTALLA DE PERSONALIZACIÓN DE LOS ENCHUFES DE LA REGLETA	- 2 -
FIGURA B-5: FRAGMENTO DE RESTABLECIMIENTO DE LA PERSONALIZACIÓN DE LOS ENCHUFES - PANTALLA DE ANALÍTICAS DE USO SOBRE CADA UNO DE LOS ENCHUFES - PANTALLA DE GENERACIÓN DE NOTIFICACIONES Y EVENTOS SOBRE LOS ENCHUFES -	- 3 -
FIGURA B-6: PANTALLA DE PRESENTACIÓN EN RELOJES INTELIGENTES – PANTALLA DE CONTROL DEL ENCHUFE 1 – PANTALLA DE ACCESO A LOS AJUSTES DE LA APLICACIÓN	- 3 -
FIGURA B-7: PANTALLA DE AJUSTES DE LA APLICACIÓN – PANTALLA DE CONEXIÓN CON LA RASPBERRY PI – PANTALLA DE SELECCIÓN DE LOS ENCHUFES	- 3 -
FIGURA B-8: PANTALLA DE PERSONALIZACIÓN DE LOS ENCHUFES - PANTALLA DE GENERACIÓN DE NOTIFICACIONES Y EVENTOS SOBRE LOS ENCHUFES	- 4 -

1 Introducción

1.1 Motivación

Un comunicador consiste en una colección de imágenes y pictogramas que ayudan a transmitir información sin leer ni utilizar la voz. Se emplea para mejorar la comunicación en niños o adultos con dificultades altas de habla o muy bajo nivel cognitivo, tales como las que pueden causar trastornos del espectro autista, deficiencias intelectuales y otros tipos de discapacidades, permitiendo desarrollar comportamientos interactivos espontáneos intencionales.

En este proyecto, se propone desarrollar una familia de comunicadores basados en aplicaciones para teléfonos, tabletas y relojes inteligentes con sistema operativo Android, que en conjunto se ajusten a diferentes necesidades de cada uno de los usuarios y puedan ser útiles para facilitar la ejecución de tareas de la vida cotidiana, así como encender o apagar cualquier elemento eléctrico de una vivienda, como una bombilla o un televisor, integrando completamente el sistema en un comunicador Android dirigido a personas con un bajo nivel cognitivo, otorgando así una mayor autonomía a personas con necesidades especiales.

La aplicación debe ofrecer, para cada una de las plataformas, un entorno sencillo, que sea intuitivo, organizado y accesible, con pocos elementos y una interfaz móvil simple. Se busca que este ecosistema sea útil para todas las personas, independientemente de su nivel cognitivo. Por lo tanto, la interfaz de usuario debe producir una estimulación sensorial, fundamentalmente a través de elementos visuales que generen atención y motivación, como imágenes y pictogramas. Su manejo debe ser simple, de manera que mitigue la frustración, y ofreciendo un refuerzo alto e inmediato, fortaleciendo la autonomía y el desarrollo de las interacciones y el autocontrol.

El proyecto es abordado por dos socios que colaboran desde 2017. Los aspectos educativos, las especificaciones funcionales, y las pruebas e implantación son llevadas a cabo por dos especialistas del CEIP Romero Peña de La Solana, en Ciudad Real, las licenciadas Juana Domínguez Fernández-Arroyo y Josefa García Ariza. Ambas especialistas están a cargo de la Unidad de Educación Especial, que atiende a los alumnos/as con problemas de audición y lenguaje tales como TEA, parálisis cerebral, síndrome de Down y síndrome de Angelman.

Complementariamente, el diseño electrónico, desarrollo, programación, test funcional, documentación, dirección técnica, soporte y gerencia, además de todos los detalles tecnológicos, son responsabilidad del Laboratorio de Sistemas Digitales de la Universidad Autónoma de Madrid.

Una de las principales contribuciones de este Trabajo de Fin de Máster es agregar al comunicador el control de una regleta de enchufes inteligentes mediante una Raspberry Pi pudiendo así gestionarlos de manera remota y efectiva analizando su uso desde una aplicación para el sistema operativo Android. Es decir, una aplicación de comunicador Android para tabletas, teléfonos y relojes inteligentes integrada con un control de enchufes de una regleta inteligente mediante la red Wifi. La aplicación tendrá una versión individual

y tendrá otra versión integrada en la aplicación QUIERO DECIRTE desarrollada en mi Trabajo de Fin de Grado [1] y disponible en la Google Play Store.

El sistema se planea con un alto nivel de personalización al usuario, con una interfaz de usuario sencilla e intuitiva, además de seguridad a cambios de estado de los enchufes, con analíticas de uso y una base de datos de registro, con la posibilidad de apagar o encender individualmente cada enchufe desde un dispositivo Android a distancia, programar notificaciones al usuario de enchufes específicos y apagar cada uno de ellos a una hora determinada.

Las verificaciones se realizan de manera continua en el CEIP Romero Peña de La Solana. Los resultados se realimentan al equipo de la UAM, que se encarga de realizar del rediseño, hasta alcanzar un nivel estable del sistema.

1.2 Objetivos

El objetivo principal de este Trabajo de Fin de Máster es integrar el control de una regleta de enchufes inteligente en una familia de comunicadores Android dirigida a personas con discapacidades tales como el TEA que provocan una disminución del nivel cognitivo, proporcionando comodidad al usuario y facilitando la realización de tareas de la vida cotidiana de las personas, tales como cargar un teléfono móvil, encender o apagar la televisión o cualquier otro dispositivo que se conecte a la red eléctrica de los domicilios o edificios de una forma responsable y eficiente, ofreciendo la opción de monitorizar el uso de cada enchufe y personalizarlo, así como también controlar el tiempo que está cada uno emitiendo corriente y poder apagar y encender cada enchufe de una regleta de manera remota, sencilla y accesible para todas las personas mediante una interfaz de usuario accesible y con pocos estímulos.

Para conseguir el objetivo principal hará falta llegar a una serie de objetivos parciales que en conjunto desarrollarán de forma eficaz el ecosistema de dispositivos de sistema operativo Android conectados a una regleta inteligente. Será necesario para ello:

- Desarrollar con éxito una aplicación accesible e intuitiva que esté soportada en dispositivos tabletas, teléfonos y relojes inteligentes con sistema operativo Android, que permita toda la personalización de los enchufes, el control y las analíticas de cada uno de ellos y la conexión mediante Wifi con la Raspberry Pi conectada a la regleta e integrarla completamente en un comunicador Android dirigido a personas con discapacidades que dificulten sus capacidades de comunicación personal y sus interacciones.
- Configurar el ordenador de placa reducida Raspberry Pi para que esté en conexión con la aplicación Android mediante Wifi y ejerza el control sobre los actuadores que conecten con la regleta conectando y desconectando cada uno de los enchufes individualmente, siendo capaz de recibir información de otros dispositivos mediante la red Wifi.
- Implementar y desarrollar el circuito actuador que conecte la Raspberry con la regleta quedando integrada en una caja no siendo visible para el usuario y consiguiendo que la funcionalidad de la aplicación se lleve a cabo en la regleta.

Uniendo todos los objetivos parciales se conseguirá llegar al objetivo principal del Trabajo de Fin de Máster.

Respecto a las características de la aplicación Android, que harán que la interfaz sea cómoda, accesible y eficiente, serán las siguientes:

- Habrá una pantalla de inicio de sesión y registro conectada a una base de datos de la herramienta Firebase para que sólo pueda configurar la regleta los usuarios permitidos o supervisores.
- Existirá la posibilidad de encender y apagar los enchufes de la regleta de manera individual o de manera conjunta remota e instantáneamente.
- Permitirá la posibilidad de bloqueo de los enchufes para que no puedan ser apagados o encendidos por usuarios que no hayan sido registrados en la aplicación sin supervisión.
- La personalización de cada uno de los enchufes será un punto importante, pudiendo establecerles un nombre y una imagen o pictograma visibles dentro de la aplicación y la posibilidad también de eliminarlos y restablecerlos para proporcionar mayor autonomía a los usuarios.
- Estará disponible una pantalla de analíticas de uso que muestre cuánto tiempo ha estado activado cada uno de los enchufes accesible para los supervisores.
- Existirá la posibilidad de recibir notificaciones en el dispositivo de cada uno de los enchufes programadas en el tiempo y la posibilidad de apagar cualquier enchufe en un tiempo determinado por el usuario supervisor.

Dichas características de la aplicación conseguirán facilitar tareas de la vida cotidiana de todos los usuarios que la dispongan, independientemente de su nivel cognitivo, aportando una mayor autonomía y facilitando el aprendizaje de personas con TEA y otras discapacidades que impliquen un nivel cognitivo bajo.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 1: Introducción
- Capítulo 2: Estado del arte
- Capítulo 3: Diseño
- Capítulo 4: Desarrollo
- Capítulo 5: Integración, pruebas y resultados.
- Capítulo 6: Conclusiones y trabajo futuro.

2 Estado del arte

2.1 Introducción

Mediante la implantación de la domótica, cada vez en más hogares, se puede lograr la automatización de viviendas produciendo una mayor seguridad para las personas y contribuyendo a una mejora en la eficiencia en el uso de elementos eléctricos. Este aumento del mundo de la domótica en edificios y viviendas se debe a los grandes avances tecnológicos de los últimos años y el deseo de invertirlos en la facilitación de la vida de los usuarios y su calidad.

Hoy en día en la mayoría de los hogares y edificios existen una gran cantidad de elementos electrónicos, que aumentan con el tiempo y que necesitan de un enchufe para poder funcionar. Por ello, actualmente, el mercado de regletas de enchufes inteligentes controlados mediante aplicaciones Android por Wifi está en auge por su simplicidad y su gran utilidad, apareciendo una serie de productos de diferentes compañías y con distintas características, siendo de gran ayuda en la facilitación de las tareas de la vida cotidiana de los usuarios.

2.2 Sistemas disponibles similares

A continuación, se pasará a comentar algunos de los sistemas de regletas de enchufes más relevantes del mercado y las aplicaciones Android que las controlan, presentando sus principales características y funcionalidades. Debido a que la gran mayoría de las regletas de enchufes del mercado utilizan la misma aplicación Android, primero se comentará por separado dichas regletas y después las aplicaciones existentes.

2.2.1 Regletas de enchufes inteligentes

En este apartado se presentarán las regletas de enchufes más relevantes del mercado y sus especificaciones y características.

2.2.1.1 Regleta Inteligente Wifi LOONFREE

La regleta inteligente de la marca LOONFREE [2], disponible en el mercado, cuenta con las siguientes características:

- Contiene seis enchufes de corriente alterna y cuatro puertos USB de 5 voltios.
- Es controlada por la aplicación *Smart Life*, la cual pasará a comentar más adelante, disponible para los sistemas operativos Android e iOS.
- La regleta puede ser controlada también mediante el asistente de voz Alexa y Google Home.
- Permite establecer un horario de apagado y encendido de la regleta y puede ser controlada desde varios dispositivos de manera simultánea.

Tras analizar todas sus especificaciones, como aspectos mejorables se han encontrado los siguientes:

- Los puertos USB no son inteligentes, es decir, una vez que la regleta esté enchufada a la corriente, funcionan siempre y no se pueden controlar.

- Los enchufes no funcionan individualmente, es decir, no se tratan de enchufes inteligentes, sino de filas inteligentes, ya que desde la aplicación solo se pueden controlar las filas de la regleta.
- Se puede controlar mediante dispositivos móviles y tabletas, pero no mediante relojes inteligentes.
- Desde los asistentes de voz simplemente se puede apagar o encender la regleta entera.
- La regleta, respecto a la conexión Wifi, solo puede ser controlada con redes de 2,4 GHz, excluyendo las redes de 5 GHz, no dando la posibilidad de obtener una conexión más veloz y con menos interferencias.



Figura 2-1: Regleta de enchufes inteligentes Wifi LOONFREE

2.2.1.2 Regleta Inteligente Wifi TAOCOCO

La regleta inteligente de la marca TAOCOCO [3], disponible también en el mercado, cuenta con las siguientes especificaciones:

- La regleta cuenta con cuatro enchufes inteligentes de corriente alterna y tres tomas USB.
- Se controla, también, mediante la aplicación *Smart Life*, disponible para Android y para iOS.
- Los enchufes de corriente alterna se pueden controlar de manera individual.
- Se pueden configurar horarios de encendido y apagado de cada uno de los enchufes mediante temporizadores.
- La regleta es compatible con los asistentes de voz Alexa, Google Home e IFTTT.

Como aspectos mejorables que se he encontrado a esta regleta inteligente destacan los siguientes:

- La conexión Wifi falla continuamente y muchos de los usuarios finalmente la tienen que utilizar como una regleta de enchufes convencional.
- Los puertos USB se pueden controlar de manera conjunta y no de forma individual.
- Es compatible con dispositivos móviles y tabletas, pero no con relojes inteligentes.

- Respecto a la conexión Wifi, solo se conecta a la red de 2,4 GHz, pero no a la red de 5 GHz, restringiendo la posibilidad de obtener una conexión más rápida y con menos interferencias.



Figura 2-2:Regleta de enchufes inteligentes Wifi TAOCOCO

2.2.2 Aplicaciones Android controladores de regletas de enchufes inteligentes

2.2.2.1 Smart Life – Smart Living

Esta aplicación está disponible para su descarga en la Google Play Store para móviles y tabletas con sistema operativo Android y en la App Store para móviles y tabletas con sistema operativo iOS.

Smart Life [4] es la aplicación que utilizan la gran mayoría de las regletas de enchufes inteligentes que existen en el mercado actualmente. Permite encender y apagar los enchufes de dichas regletas y se puede utilizar desde varios dispositivos simultáneamente.

Las características y funcionalidades de esta aplicación son las siguientes:

- Permite controlar y gestionar dispositivos inteligentes de varias marcas diferentes, pudiendo encender, apagar o regular cualquier dispositivo eléctrico conectados a los enchufes registrados.
- Es compatible con los asistentes de voz de Alexa, Google Home e IFTTT.
- Permite que varios dispositivos gestionen los enchufes inteligentes de manera simultánea.
- Tiene una gran cantidad de ejemplos de tipos de dispositivos electrónicos para diferenciar los enchufes y permite ponerles nombre.
- Permite establecer horarios para apagar o encender los dispositivos en momentos determinados por el usuario y también según escenarios ambientales, como la temperatura o la humedad de la ciudad en la que vive el usuario.

Tras analizar la aplicación, se han encontrado los siguientes aspectos mejorables:

- La interfaz de usuario es poco intuitiva y podría ser más sencilla de utilizar.

- La personalización de los enchufes podría ser mayor, ya que sólo permite incluir los ejemplos que la aplicación tiene predeterminados y no se puede añadir imágenes a dichos enchufes para diferenciarlos.
- No ofrece analíticas sobre cuánto tiempo se utiliza cada dispositivo ni permite programar avisos de tiempo de los enchufes para que manden notificaciones al usuario.
- No permite la conexión Wifi de 5 GHz, simplemente la de 2,4 GHz, no ofreciendo una conexión más rápida y con menos interferencia.
- Para utilizar la aplicación en varios dispositivos de manera simultánea, todos los usuarios deben compartir la misma cuenta y contraseña.
- Para acceder a todas las funcionalidades de la aplicación, el usuario debe aceptar que *Smart Life* se inicie cuando el dispositivo se encienda automáticamente.
- No está disponible para relojes inteligentes.

La aplicación cuenta con más de un millón de descargas en la Google Play Store de Android y fue creada por la compañía desarrolladora de software Tuya Inc.



Figura 2-3: Logotipo Smart Life – Smart Living

2.2.2.2 Kasa Smart

Esta aplicación también está disponible para los sistemas operativos Android, desde la Google Play Store, y para iOS, desde la App Store, para dispositivos móviles y tabletas.

Kasa Smart [5] es capaz de controlar y gestionar los productos domésticos inteligentes de la marca TP-Link que actualmente existen en el mercado, conectándose directamente con ellos.

La aplicación cuenta con las siguientes características y especificaciones:

- Permite apagar, encender y regular dispositivos inteligentes de la marca TP-Link.
- Es compatible con los asistentes de voz Alexa, IFTTT y Google Home.
- Cuenta con un modo seguro que apaga y enciende dispositivos de manera aleatoria cuando no hay ningún usuario en casa para dar la sensación de que la vivienda no está vacía.
- Dispone de una lista de dispositivos, ordenada en categorías, que pueden ser conectados a la aplicación y permite ponerles un nombre para distinguirlos.
- Permite establecer horarios y escenarios para que el sistema apague o encienda algún dispositivo conectado.

Tras analizar las funcionalidades de la aplicación, se recalcan los siguientes aspectos que pueden ser mejorados:

- La personalización de los enchufes podría ser mejor, ya que sólo puedes conectar una pequeña lista de dispositivos y cambiarles el nombre, pero no añadirles una imagen o pictograma para diferenciarlos.
- Tampoco ofrece la posibilidad de establecer la conexión con los dispositivos mediante la red 5 GHz de Wifi, simplemente mediante la red de 2,4 GHz.
- No ofrece analíticas sobre cuánto tiempo se ha utilizado cada uno de los dispositivos, ni ofrece notificaciones al usuario sobre los mismos.
- No está disponible para relojes inteligentes.
- La aplicación cuenta con varias funcionalidades que están sin desarrollar.

La aplicación cuenta con más de un millón de descargas en la Google Play Store y ha sido desarrollada por la corporación TP-LINK Research America.



Figura 2-4: Logotipo Kasa Smart

La gran mayoría de las regletas actuales y aplicaciones del mercado no permiten la personalización de los enchufes mediante nombres e imágenes. Tampoco permiten mandar notificaciones al usuario de avisos de enchufes, además de no proporcionan datos analíticos del tiempo de uso de cada uno de los enchufes, no permiten el bloqueo de los enchufes y el usuario no tiene por qué registrarse en una base de datos interna de la aplicación para mejorar la seguridad. Estos aspectos se implementarán también en este Trabajo de Fin Máster mejorando así algunas de las características de las regletas inteligentes y aplicaciones que existen en el mercado actual.

3 Diseño

3.1 Objetivos de diseño

Este trabajo de fin de máster tiene como objetivo desarrollar una aplicación para dispositivos con sistema operativo Android, ya sean como teléfonos, tabletas y relojes inteligentes, capaz de controlar y gestionar una regleta de enchufes inteligente conectados a una Raspberry Pi, un ordenador de placa reducida, mediante la red Wifi e integrarlo completamente en el comunicador *Quiero Decirte*.

La aplicación tendrá una versión que funcionará de manera individual en los dispositivos que lo deseen y otra versión que irá integrada completamente en la aplicación *Quiero Decirte*, desarrollada previamente en mi Trabajo de Fin de Grado, que hace la función de comunicador mediante imágenes, pictogramas y locuciones, dirigida a personas con un nivel cognitivo bajo y dificultades de comunicación tales como personas con trastorno del espectro autista o deficiencias intelectuales.

Por ello, como la aplicación irá dirigida a todos los usuarios independientemente de su nivel cognitivo, uno de los aspectos más destacables en su desarrollo es presentar una interfaz de usuario sencilla, con pocos estímulos, además de intuitiva, accesible para todos los usuarios y predecible.

Respecto a la conexión de la aplicación con la regleta de enchufes inteligentes mediante la Raspberry Pi, deberá ser instantánea y segura, pudiendo ofrecer una serie de ajustes y facilidades al usuario para poder conectar cualquiera de los enchufes de una forma cómoda y sencilla.

3.2 Requisitos de diseño de la aplicación Android

Para el acceso de la aplicación y su ejecución es necesario establecer una serie de requerimientos y especificaciones que serán comentados a continuación.

3.2.1 Especificaciones técnicas

Las especificaciones técnicas son las que definen qué dispositivos son capaces de ejecutar la aplicación y cómo lo harán, tales como el sistema operativo, su versión, el tamaño de las pantallas y los dispositivos permitidos.

3.2.1.1 Sistema Operativo

Para la selección del sistema operativo de la aplicación se realizó un estudio de mercado para poder desarrollarla y que pueda llegar al máximo número de usuarios y dispositivos, teniendo en cuenta que la aplicación deberá estar disponible para teléfonos, tabletas y relojes inteligentes.

Analizando el mercado se puede observar las siguientes gráficas, generadas mediante la plataforma de analíticas de StatCounter [6], que muestran los porcentajes de la utilización de los diferentes tipos de dispositivos que albergarán la aplicación y su conjunto dependiendo de su sistema operativo.

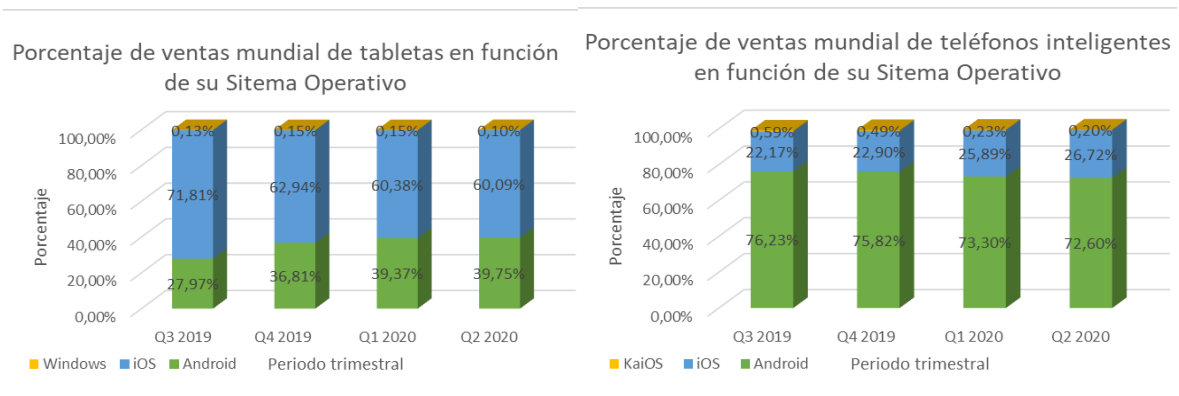


Figura 3-1: Tablet y teléfonos más vendidos según su sistema operativo

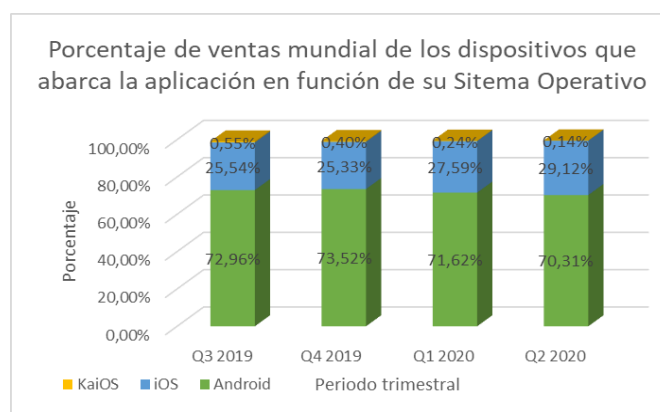


Figura 3-2: Ventas de los dispositivos que abarca el proyecto más vendidos según su sistema operativo

De este estudio de mercado se puede apreciar que el líder de ventas de dispositivos es el sistema operativo Android, ya que juntamente con todos los dispositivos tiene un volumen de ventas muchísimo mayor que los demás sistemas operativos. Salvo en el caso individual de las tabletas, que el líder en el mercado es el sistema operativo iOS; aunque Android poco a poco va aumentando sus ventas mientras que el sistema operativo de Apple las va disminuyendo.

También un punto importante a la hora de seleccionar el sistema operativo es la accesibilidad y comodidades a la programación que ofrecen cada uno de ellos. Android ofrece una gran cantidad de facilidades a los programadores, así como un gran número de herramientas gratuitas de desarrollo, pruebas con dispositivos virtuales, una gran cantidad de documentación sobre todas sus funciones, además de código abierto con ejemplos para entender su funcionamiento y para obtener una cuenta de desarrollador y poder incluir aplicaciones a la Google Play Store únicamente hay que realizar un pago de 25 dólares sin límite de tiempo. Por otro lado, el sistema operativo iOS ofrece herramientas software para el desarrollo de aplicaciones de pago, obtener dispositivos físicos para realizar pruebas reales es más costoso y obtener una cuenta de desarrollador para introducir aplicaciones en la App Store tiene un precio de 99 dólares anuales.

Una vez elegido el sistema operativo Android por todos estos argumentos, es necesario establecer para qué versiones estará disponible la aplicación buscando el equilibrio entre

poder llegar al máximo número de usuarios y que el dispositivo sea capaz de realizar todas las funcionalidades durante su ejecución.

El porcentaje de dispositivos Android que soportan las diferentes versiones del sistema operativo se ve reflejado en el siguiente gráfico de manera acumulativa [7].

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,8%
4.2 Jelly Bean	17	99,2%
4.3 Jelly Bean	18	98,4%
4.4 KitKat	19	98,1%
5.0 Lollipop	21	94,1%
5.1 Lollipop	22	92,3%
6.0 Marshmallow	23	84,9%
7.0 Nougat	24	73,7%
7.1 Nougat	25	66,2%
8.0 Oreo	26	60,8%
8.1 Oreo	27	53,5%
9.0 Pie	28	39,5%
10. Android 10	29	8,2%

Figura 3-3: Porcentaje de las versiones de Android según los dispositivos que las soportan

Para el desarrollo de la aplicación Android se ha optado por utilizar la versión Android 4.4 KitKat, ya que es una versión del sistema operativo que permite realizar todas las funcionalidades del proyecto sin ningún tipo de problema y la gran mayoría de los dispositivos del mercado actual son compatibles.

3.2.1.2 Diseño de pantallas

Las especificaciones de este Trabajo de Fin de Máster requieren que la aplicación sea compatible con distintos dispositivos de diferentes tamaños de pantalla, como teléfonos, tabletas y relojes inteligentes. Por ello, es necesario tener en cuenta el rango de sus dimensiones de pantalla y sus densidades de píxeles para diseñar sus interfaces de manera óptima.

El sistema operativo Android ofrece la posibilidad a los desarrolladores de adaptar la ejecución de sus aplicaciones a diferentes dispositivos con diferentes pantallas mediante las herramientas de diseños lineales, donde los elementos de la interfaz de usuario tales como botones y textos quedan alineados ajustándose a las pantallas y sus densidades.

Las pantallas de los dispositivos tienen dos características fundamentales; el tamaño físico de la pantalla, generalmente en pulgadas, y la densidad de píxeles (dpi). El tamaño físico de la pantalla establece el espacio real que ocupa la aplicación en la zona del dispositivo, mientras que la densidad de píxeles mide el número de puntos de píxel por pulgada de la pantalla. En la siguiente imagen aparece una comparativa sobre los diferentes tamaños de pantalla y sus densidades, categorizados según rangos [8].

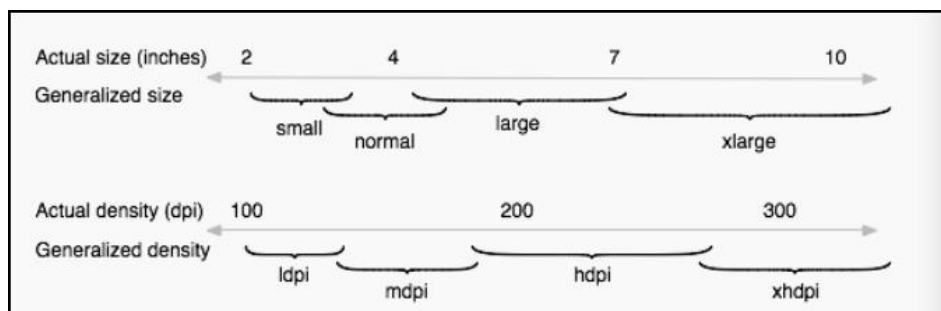


Figura 3-4: Comparativa entre tamaños de pantalla y densidades de píxeles

En este Trabajo de Fin de Máster, como se abarcarán los dispositivos móviles, tabletas y relojes inteligentes, se trabajará con todos los rangos de pulgadas de la imagen y todos los rangos de densidades de píxeles.

Respecto al diseño de la interfaz de usuario de las pantallas, la apariencia será simple e intuitiva, ya que esta aplicación irá dirigida a cualquier tipo de usuario independientemente de su nivel cognitivo e irá completamente integrada en un comunicador especializado en personas con ciertas discapacidades como el TEA o deficiencia intelectual que conllevan dificultades de comunicación y deterioro en las relaciones interpersonales. Por ello contará con pocos elementos visuales y distracciones, con un entorno útil, favorable, intuitivo y personalizable consiguiendo que la funcionalidad principal de la aplicación, que consiste en controlar una regleta de enchufes inteligentes, sea lo más sencilla posible de realizar para el usuario.

3.2.2 Estructura de la aplicación

La estructura de la aplicación será accesible y cómoda, para que sea sencillo utilizar cada una de las pantallas y que no pueda existir duda alguna sobre cada uno de los apartados que la conformen.

Debido a los diferentes tamaños de pantallas que abarca la aplicación, se han propuesto dos estructuras de aplicación diferentes para dinamizar su uso y poder adaptarse a las necesidades de cada uno de los dispositivos, diferenciando una versión para los relojes inteligentes y otra para los teléfonos y tabletas las cuales irán completamente integradas en el comunicador *Quiero Decirte* y serán accesibles de manera sencilla al usuario.

3.2.2.1 Versión para teléfonos y tabletas

Las pantallas y fragmentos de cuadros navegables de la aplicación disponible para teléfonos y tabletas siguen la siguiente estructura. Cada una de las pantallas y sus funcionalidades serán comentadas más adelante.

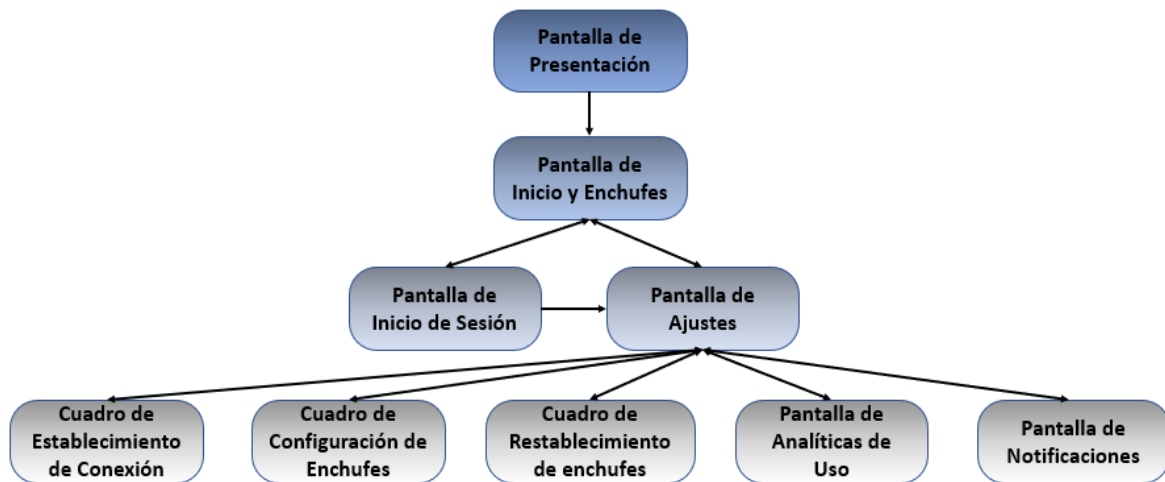


Figura 3-5: Estructura de la aplicación en la versión para teléfonos y tabletas

3.2.2.2 Versión para relojes inteligentes

Las pantallas navegables de la aplicación disponible para relojes inteligentes siguen la siguiente estructura. Cada una de las pantallas y sus funcionalidades serán comentadas, también, más adelante.

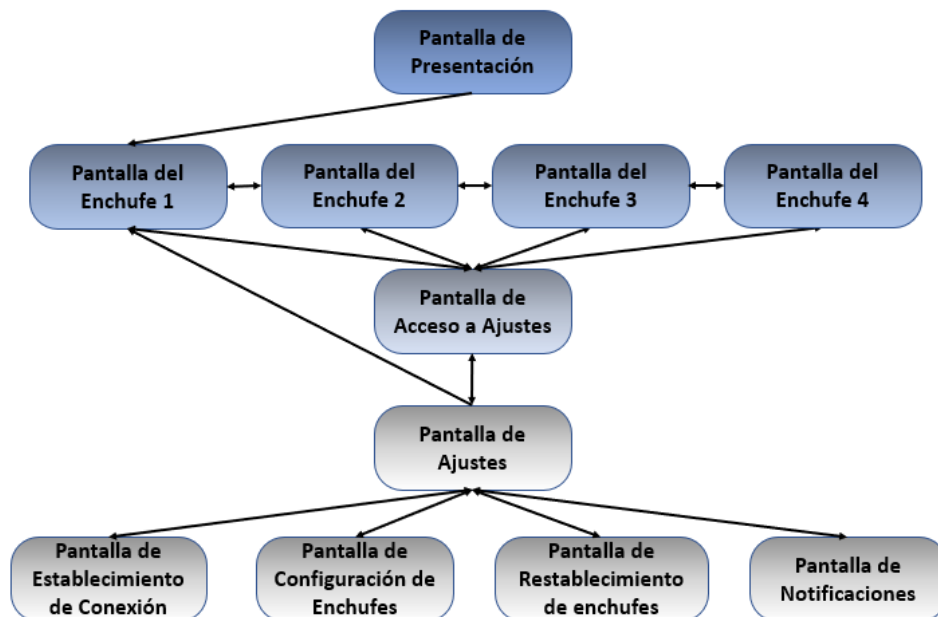


Figura 3-6: Estructura de la aplicación en la versión para relojes inteligentes

3.3 Requisitos del diseño hardware

La regleta de enchufes deberá ir conectada a un microprocesador o microcontrolador que sea capaz de recibir datos y órdenes por parte de la aplicación Android mediante teclados. Inicialmente es indispensable establecer mediante qué red la aplicación enviará los datos a la regleta estudiando las principales conexiones disponibles en todos los tipos de dispositivos que abarcará este Trabajo de Fin de Máster para después decidir qué microprocesador o microcontrolador será el idóneo para llevar a cabo el objetivo.

Actualmente, prácticamente todos los teléfonos, tabletas y relojes inteligentes disponen de dos de los tipos de redes más utilizados para enviar datos, la red Wifi y la red Bluetooth. Para seleccionar uno de estos dos tipos de conexiones se analizan las características de cada uno más relevantes para desarrollar este proyecto:

- Bluetooth [9]: su frecuencia es de 2,4 GHz, su ancho de banda es de 24 Mbps, cuenta con una seguridad baja, su rango es de hasta 30 metros y el consumo que ejerce en los dispositivos es bastante bajo.
- Wifi [10]: su frecuencia es de 2,4 GHz, 3,6 GHz y 5 GHz, su ancho de banda es de 1 Gbps, su seguridad es moderada, su rango es de hasta 300 metros pudiendo incorporar repetidores y tiene un consumo elevado en los dispositivos.

Por todas estas características finalmente se decide emplear la red Wifi para realizar la conexión entre los dispositivos Android y la regleta de enchufes inteligentes, ya que establece una conexión más segura, mucho más rápida y con mucha más distancia entre dispositivo y enchufes. Por contrapunto tiene un consumo mayor, pero como la conexión sólo se establecerá cuando se encienda o apague un enchufe y al establecer la conexión, no supondrá ningún problema en el desarrollo de su funcionalidad.

Una vez que está establecida la red Wifi como conexión entre regleta y dispositivos Android, el siguiente punto es decidir qué dispositivo hará la función de receptor para recoger los telemandos y actuar sobre los enchufes inteligentes de la regleta. Para ello se realiza un estudio entre microprocesadores y microcontroladores del mercado actual analizando sus características:

- Arduino [11]: es una compañía de desarrollo de hardware y software libre que también manufactura placas de desarrollo de hardware para crear dispositivos que pueden ser conectados a elementos electrónicos de la vida cotidiana. Cuenta con una familia de microcontroladores con diferentes características y funcionalidades. Existen pocas placas que cuenten con Wifi integrado, pero existen módulos para incluirlos, aumentando el precio y disminuyendo el número de puertos disponibles. Los microcontroladores Arduino que más se adecuan a las funcionalidades de este Trabajo de Fin de Máster son los siguientes:
 - Arduino MKR WIFI 1010 [12]: cuenta con un módulo Wifi de 2,4 GHz, 8 pines digitales que funcionan a 3,3 voltios y 12 pines PWM, una memoria flash de 256 KB y una SRAM de 32 KB.
 - Arduino UNO WIFI Rev 2 [13]: cuenta con un módulo Wifi de 2,4 GHz, 14 pines digitales que funcionan a 5 voltios y 6 puertos analógicos, una memoria flash de 32 KB, una SRAM de 2 KB y una EEPROM de 1KB.

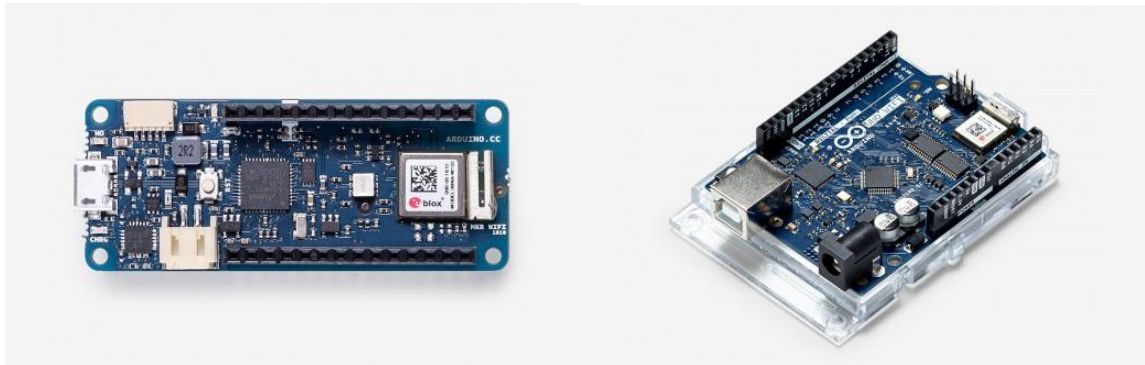


Figura 3-7: Productos Arduino: Arduino MKR WIFI 1010 – Arduino UNO Wifi Rev 2

- Raspberry Pi [14]: es un ordenador de placa reducida de bajo coste, pero con una gran potencia. Cuenta con su propio sistema operativo de código abierto, pero permite instalar otros sistemas tales como Linux o Windows 10. Permite desarrollar programas complejos y cuenta con una gran cantidad de funcionalidades. Los ordenadores de placa reducida Raspberry Pi que se adecuan más a las funcionalidades de este proyecto son los siguientes:
 - Raspberry Pi 3 Modelo a+ [15]: cuenta con una memoria SDRAM de 512 MB y una frecuencia de trabajo de 1,4 GHz, dispone de 40 pines GPIO, una entrada HDMI para realizar la programación, un único puerto USB 2.0, un módulo Wifi de 2,4 GHz y 5 GHz y otro Bluetooth 4.2.
 - Raspberry Pi 3 Modelo b+ [16]: cuenta con una memoria SDRAM de 1 GB y una frecuencia de trabajo de 1,4 GHz, dispone de 40 pines GPIO, una entrada HDMI, cuatro puertos USB 2.0, un módulo Wifi de 2,4 GHz y 5 GHz y otro Bluetooth 4.2.
 - Raspberry Pi 3 Modelo b [17]: cuenta con una memoria RAM de 1 GB y una frecuencia de trabajo de 1,2 GHz, dispone de 40 pines GPIO, una entrada HDMI, cuatro puertos USB 2.0, un módulo Wifi de 2,4 GHz y otro Bluetooth BLE.



Figura 3-8: Productos Raspberry Pi: Raspberry Pi 3 Modelo 1+ - Raspberry Pi 3 Modelo b+ - Raspberry Pi 3 Modelo b

Debido a la reducida cantidad de puertos de entrada y salida que ofrecen los microcontroladores de Arduino con módulo Wifi, su poca memoria, velocidad y que no disponen de conexión a la red Wifi de 5 GHz me decanto por los ordenadores de placa reducida Raspberry Pi.

Una característica relevante del proyecto es la velocidad con la que debe trabajar el hardware para recibir los datos desde la aplicación Android y poder actuar sobre los

enchufes de la regleta, por ello descarto la Raspberry Pi 3 Modelo b, por su menor frecuencia de trabajo y por no ser capaz de conectarse a la red Wifi de 5 GHz. Respecto a las diferencias entre la Raspberry Pi 3 Modelo a+ y b+, la versión a+ cuenta con una menor memoria SDRAM y un único puerto USB, indispensables para poder conectar ratón y teclado para llevar a cabo toda la programación y navegar por la interfaz del sistema operativo de la placa de una forma más cómoda y sencilla. Por todo ello me decanto por la Raspberry Pi 3 Modelo b+ para recibir los datos de la aplicación Android y actuar sobre los enchufes de la regleta inteligente.

4 Desarrollo

En este apartado pasaré a comentar todos los aspectos importantes del desarrollo del proyecto, explicando cada uno de los siguientes puntos:

- Herramientas aplicadas.
- Desarrollo de la aplicación Android.
 - Versión para móviles y tabletas.
 - Versión para relojes inteligentes.
 - Implementación en aplicación *Quiero Decirte*.
- Desarrollo de la regleta de enchufes inteligentes.
 - Elementos de la regleta de enchufes inteligentes.
 - Diseño del sistema.
- Desarrollo de la conexión entre la aplicación Android y la regleta de enchufes inteligentes.

4.1 Herramientas aplicadas

Para la realización de este proyecto han sido necesarias los siguientes elementos y herramientas:

- Ordenador portátil.
- Software Android Studio versión 3.6.2 para el desarrollo de la aplicación Android que establecerá la conexión con la regleta de enchufes inteligente.
- Última versión de la aplicación comunicador *Quiero Decirte*.
- Teléfono y tableta para la realización de pruebas del proyecto.
- Raspberry Pi 3 Modelo b+.
- Tarjeta Micro SD con el instalador de sistemas operativos NOOBS (New Out of the Box Software).
- Pantalla y cable HDMI para ver la imagen del ordenador de placa reducida.
- Ratón y teclado para interactuar con la Raspberry Pi 3 Modelo b+.
- Protoboard.
- Piezas de cable hembra-macho y hembra-hembra.
- Leds y resistencias.
- Módulo de relé de 4 canales.
- Bases de enchufes modulares.
- Cable de instalación eléctrica.

4.2 Desarrollo de la aplicación Android

La aplicación ha sido desarrollada en dos versiones diferentes para adaptarse a las pantallas y características de los diferentes dispositivos que serán capaces de ejecutarla, existiendo una versión para teléfonos y tabletas y otra para relojes inteligentes, debido a su reducido tamaño, ambas completamente integradas en el comunicador *Quiero Decirte*.

El nombre de la aplicación es *Remote Control* y a continuación comentaré su funcionamiento y desarrollo diferenciando las dos versiones disponibles y su posterior integración en el comunicador Android.



Figura 4-1 : Logotipo de la aplicación Control Remoto

4.2.1 Versión para móviles y tabletas

La versión de la aplicación ha sido desarrollada íntegramente en el software Android Studio mediante los lenguajes de programación Java y XML. Cuenta con un total de 11 ficheros Java con 2600 líneas de código donde se desarrollan todas las funcionalidades de la aplicación y un total de 20 ficheros XML con 2000 líneas de código donde se desarrollan todas las interfaces de las vistas y actividades donde interactuarán los usuarios.

Para desarrollar las funcionalidades del proyecto se ha generado la aplicación, mediante la programación de códigos Java y XML, de la siguiente manera:

- Al iniciar la aplicación se ha diseñado una pantalla de presentación donde aparece el logotipo y el nombre de la aplicación, se ha programado para que aparezca durante tres segundos y después de paso a la pantalla principal.

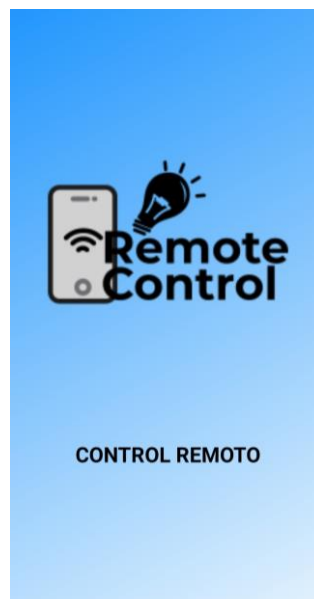


Figura 4-2: Pantalla de presentación de la aplicación

- La pantalla principal es donde el usuario pasará el mayor tiempo activo en la aplicación. En cuanto a su diseño, se ha desarrollado una interfaz sencilla e intuitiva, ya que esta pantalla será accesible para todos los usuarios independientemente de su nivel cognitivo, donde aparece la lista de enchufes de la regleta con su imagen y nombre correspondiente una vez que se ha establecido la conexión con la Raspberry Pi y se han personalizado los enchufes individualmente. Cada enchufe cuenta con un *switch* que encenderá o apagará

el usuario mandando los datos a la Raspberry Pi actuando sobre la regleta inteligente, apareciendo un mensaje en la pantalla informando sobre qué acción se ha realizado al cambiar sus estados.

También se ha diseñado una *toolbar* (barra de herramientas superior) con un botón que traslada al usuario a la pantalla de ajustes o a la de inicio de sesión y un menú lateral, *Navigation Drawer*, donde se encuentran todas las secciones y utilidades de la aplicación, a la que también se puede acceder deslizando el dedo por la pantalla de izquierda a derecha.

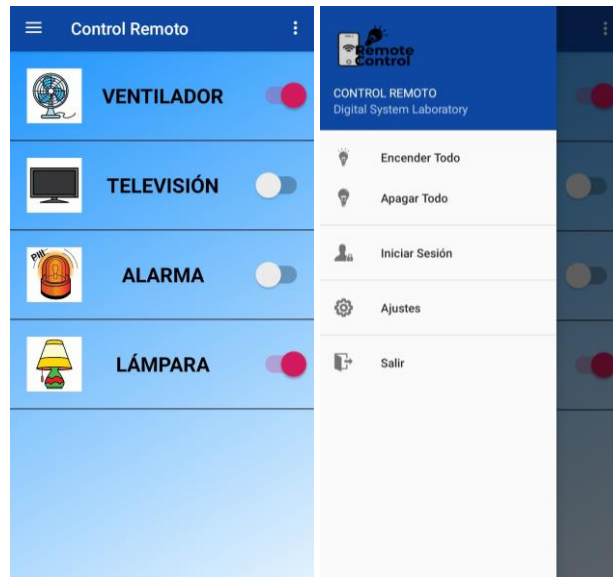


Figura 4-3: Pantalla principal de la aplicación y su menú lateral

En la parte superior del menú lateral aparece toda la información de la aplicación, como el logotipo, el nombre y los datos de contacto, al presionar la zona. En el menú aparecen unas opciones útiles para el usuario, como las opciones de encender o apagar todos los enchufes de forma instantánea, la opción de acceder a la pantalla de registro e inicio de sesión de la base de datos para acceder y desbloquear la pantalla de los ajustes de la aplicación, la opción de acceder a la configuración y finalmente la opción de salir de la aplicación y cerrarla del segundo plano para optimizar el rendimiento del dispositivo.



Figura 4-4: Fragmento de la información de la aplicación

- La siguiente pantalla a la que se deberá acceder en la aplicación para poder establecer la conexión, configurar y gestionar los enchufes es la pantalla de inicio de sesión y registro del usuario supervisor. Se accede desde el menú lateral de la pantalla principal y se ha diseñado de tal forma que aparezca el logotipo de la aplicación, se pueda volver a la pantalla principal mediante la flecha de regreso, y el usuario pueda introducir su correo electrónico y su contraseña para registrarse en la aplicación o iniciar sesión si ya ha sido registrado previamente. La aplicación ha sido desarrollada para que el usuario deba introducir una cuenta de correo electrónico válida y se pueda verificar el correo enviando un mensaje.

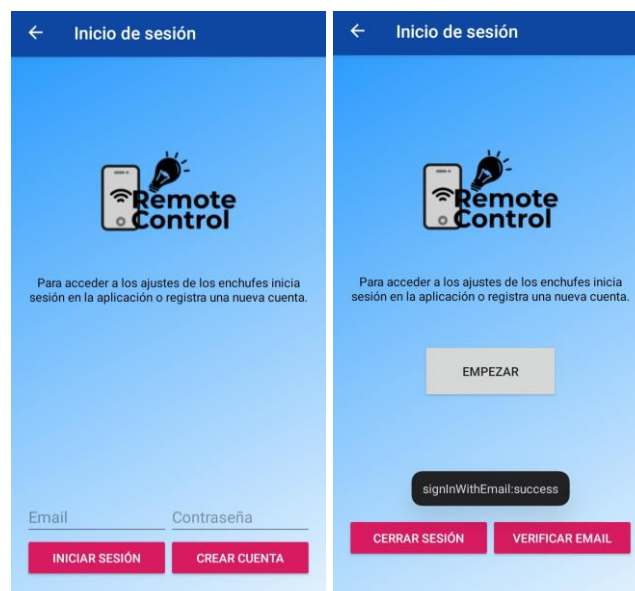


Figura 4-5: Pantalla de inicio de sesión de la aplicación

Esta pantalla está enlazada con la plataforma de desarrollo de aplicaciones de Google Firebase que cuenta con analíticas de uso, bases de datos y gestión de

usuarios. Esta herramienta controlará que solo puedan configurar los enchufes los usuarios supervisores permitidos mediante una base de datos de usuarios donde cada uno de ellos se irá registrando, ofreciendo así un gran nivel de control y seguridad a la funcionalidad del proyecto.

- Una vez que el usuario se ha identificado en la base de datos enlazada a la aplicación podrá acceder a la pantalla de los ajustes y cerrar sesión desde la pantalla principal. Esta pantalla cuenta con las siguientes categorías:

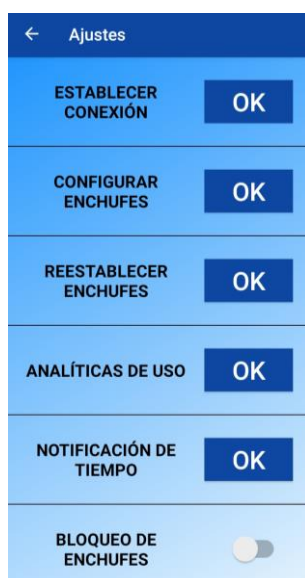


Figura 4-6: Pantalla de ajustes de la aplicación

- Un apartado de establecimiento de conexión con la placa controladora de los enchufes, en ella el usuario deberá introducir la dirección IP de la Raspberry Pi por la que se ha conectado a la red Wifi correspondiente, siendo accesible desde la configuración del router del usuario. Toda la información de la conexión entre los dispositivos Android y la Raspberry serán desarrollada más adelante.



Figura 4-7: Fragmento para la conexión con la Raspberry Pi

- Acceso a una pantalla de configuración de enchufes donde el usuario podrá personalizar cada uno de ellos individualmente. En esta pantalla los enchufes están diferenciados por colores según el led de la regleta y se deberá elegir cuál se desea configurar mediante botones circulares que permiten seleccionar una única opción, introducir un nombre y seleccionar una imagen con la cual la aplicación trabajará como mapas de bits para poder utilizarla en sus distintas pantallas. La imagen podrá ser seleccionada desde la propia galería o directamente desde la cámara del dispositivo a la vez que se solicitará permiso en ejecución para acceder a la memoria interna del dispositivo y a la cámara. El usuario sólo podrá personalizar el enchufe una vez que se haya establecido la conexión con la Raspberry Pi.

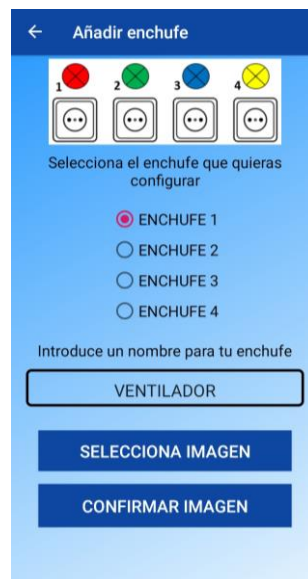


Figura 4-8: Pantalla de personalización de enchufes

- Un apartado de restablecimiento de los enchufes, donde al acceder la aplicación preguntará al usuario qué enchufe quiere desconfigurar, borrando de la aplicación su nombre, su imagen y su mapa de bits asociado.



Figura 4-9: Fragmento de selección del enchufe a restablecer

- Acceso a una pantalla de analíticas de uso, donde el usuario supervisor podrá visualizar los datos de cuánto tiempo ha estado cada enchufe encendido en días, horas, minutos y segundos. El cálculo del tiempo de uso total de cada enchufe se realiza desde la pantalla principal almacenando los momentos de apagado y encendido. Estos datos se almacenan en preferencias compartidas internas de la aplicación para que sean accesibles desde todas las pantallas. Cada enchufe también cuenta con un botón de restablecimiento de tiempo de uso.



Figura 4-10: Pantalla de las analíticas de los enchufes

- Acceso a una pantalla de notificaciones de los enchufes, donde el usuario podrá seleccionar sobre cuál quiere establecer eventos mediante un menú deslizable, podrá establecer en cuánto tiempo programará el evento introduciendo un número de días, horas y minutos y finalmente establecer qué tipo de evento desea lanzar, si recibir una notificación en el móvil recordando una acción sobre el enchufe especificado o si directamente quiere desactivarlo. Este establecimiento de eventos se desarrolla mediante una serie de servicios Android [18] que funcionan en segundo plano y cuando llega el momento establecido realizan su función, aunque la aplicación haya sido cerrada.

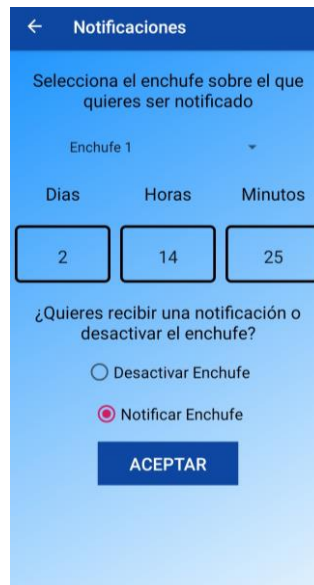


Figura 4-11: Pantalla de generación de eventos y analíticas

- Por último, la configuración de la aplicación permite el bloqueo de los enchufes en su estado actual, para ofrecer más seguridad a la regleta inteligente respecto a cambios no deseados por parte de otros usuarios no registrados en la aplicación deshabilitando la opción de encendido y apagado.

Las clases desarrolladas para que esta aplicación cumpla con todas sus funcionalidades son las siguientes:

- MainActivity.java: clase que muestra la pantalla de presentación.
- HomeActivity.java: clase que presenta la pantalla principal donde se da acceso a las demás pantallas y permite al usuario controlar los enchufes de la regleta inteligente de manera individual.
- LoginActivity.java: clase que permite al usuario registrarse en la base de datos de Firebase e iniciar sesión con sus datos, así como recibir correos de verificación, para acceder a los ajustes de los enchufes.
- Globals.java: clase que establece e inicializa todos los parámetros globales.
- User.java: clase que establece y almacena los datos de los usuarios al iniciar sesión y registrarse en la base de datos.
- BaseActivity.java: clase que gestiona los elementos de la interfaz tales como diálogos y procesos internos.
- AjustesActivity.java: clase que hospeda la interfaz de toda la configuración y personalización de la aplicación.
- AddEnchufeActivity.java: clase que permite al usuario personalizar cada uno de los enchufes cuando la conexión ha sido establecida.
- AnaliticasActivity.java: clase que permite al usuario obtener información sobre cuánto tiempo ha estado activo cada uno de los enchufes.
- NotificationActivity.java: clase que permite al usuario establecer eventos útiles sobre cada uno de los enchufes para recibir notificaciones o desactivarlos en momentos determinados.
- AlarmReceiver.java: clase que recibe los datos introducidos por el usuario sobre la programación de eventos y se encarga de lanzarlos en el momento establecido, aunque la aplicación se haya cerrado previamente.

4.2.2 Versión para relojes inteligentes

La versión de la aplicación ha sido desarrollada íntegramente en el software Android Studio y cuenta con 11 ficheros Java con 1400 líneas de código donde se ha programado todas las funcionalidades de la aplicación, y con 10 ficheros XML con 1200 líneas de código donde se ha programado toda la interfaz de usuario y el diseño de las pantallas.

Para realizar las funcionalidades de la aplicación se ha desarrollado esta versión para relojes inteligentes adaptándose a su pequeño tamaño y características mediante la programación de códigos Java y XML de la siguiente manera:

- Al iniciar la aplicación aparece en el dispositivo una pantalla de presentación donde aparece el nombre de la aplicación y el logotipo que, tras aparecer durante tres segundos, da paso automáticamente a la pantalla de control del enchufe 1.

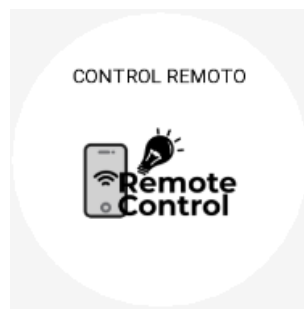


Figura 4-12: Pantalla de presentación

- La pantalla principal de la versión de teléfonos y tabletas donde se podían controlar los enchufes en la versión para relojes inteligentes está dividida en cuatro pantallas, una para controlar cada enchufe, debido al tamaño de pantalla. Estas pantallas son accesibles entre ellas deslizando el dedo de izquierda a derecha y viceversa. En cada una de ellas aparece arriba un botón que da acceso a los ajustes de la aplicación, la imagen o pictograma del enchufe establecido por el usuario y el nombre del enchufe. Para encender o apagar el enchufe basta con pulsar sobre la imagen establecida y la aplicación mandará los datos por Wifi a la Raspberry Pi.



Figura 4-13: Pantalla de control del enchufe 1

- Al presionar el botón de ajustes disponible en cada una de las pantallas de los enchufes aparecerá una pantalla de control de acceso a la configuración, donde el usuario deberá introducir una clave que será proporcionada en la información de la aplicación de la plataforma de distribución y en su manual de

instrucciones, para que no todos los usuarios puedan acceder a la configuración, aportando seguridad y control sobre la regleta de enchufes inteligente.

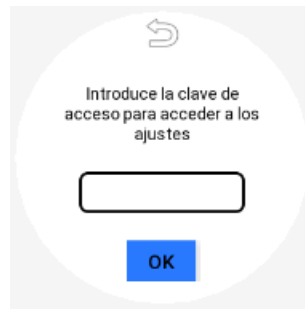


Figura 4-14: Pantalla de acceso a los ajustes de la aplicación

- Una vez que se ha introducido correctamente la clave de acceso de los ajustes de la aplicación, el usuario puede configurar todas las funcionalidades. En esta pantalla aparece una lista con los siguientes ajustes:



Figura 4-15: Pantalla de ajustes de la aplicación

- La primera opción es la de establecer la conexión con la Raspberry Pi. La aplicación pedirá al usuario que introduzca la dirección IP de la placa por la que se ha conectado a la red Wifi, disponible en la información en red del router. La conexión entre el dispositivo Android y la Raspberry Pi se comentará más adelante.



Figura 4-16: Pantalla de conexión con la Raspberry Pi

- La siguiente opción es la de configuración de los enchufes. Al acceder, la aplicación pedirá al usuario seleccionar cuál quiere personalizar mediante una serie de botones junto con una imagen donde aparecen los enchufes con su color correspondiente según su led. Una vez que se ha seleccionado el enchufe a personalizar, se deberá introducir el nombre y su imagen o

pictograma desde la galería o cámara si el dispositivo la dispone, a la vez que solicitará permiso en ejecución para acceder a la memoria interna del dispositivo y a la cámara, convirtiendo dicha imagen en un mapa de bits para poder utilizarla en las actividades de la aplicación.

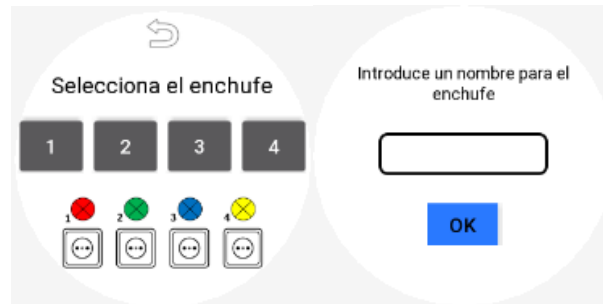


Figura 4-17: Pantallas de personalización de los enchufes

- El siguiente apartado es el de restablecimiento de un enchufe, donde el usuario podrá eliminar su configuración, así como el nombre y el mapa de bits asociado, de cada uno de ellos, accediendo a la misma pantalla de selección del apartado anterior.
- También aparece la opción de crear eventos sobre los enchufes, donde la aplicación requerirá al usuario seleccionar sobre cuál aplicarlo mediante la misma pantalla de selección que en los dos últimos apartados para después dar paso a otra pantalla donde establecer dicho evento. En esta pantalla el usuario deberá seleccionar si prefiere recibir un aviso en forma de notificación sobre el enchufe deseado o si directamente prefiere apagarlo, estableciendo un tiempo en forma de días, horas y minutos para que ocurra el evento. Una vez que se ha establecido, la aplicación generará un servicio que funciona en segundo plano, aunque no esté en ejecución, para que en el momento establecido el dispositivo Android lance su funcionalidad de manera automática.



Figura 4-18: Pantalla de generación de eventos y notificaciones

- La última opción de los ajustes de la aplicación es el bloqueo de enchufes mediante un *switch*, haciendo que el estado de los enchufes de la regleta no pueda cambiar proporcionando así seguridad a los elementos electrónicos enchufados a la regleta y evitando cambios no deseados.

Las clases programadas para que esta versión de la aplicación desarrolle todas las funcionalidades del proyecto son las siguientes:

- MainActivity.java: contiene el código de la pantalla de presentación.
- Enchufe1.java: es la clase que gestiona el control sobre el enchufe 1 y da acceso a los ajustes de la aplicación.
- Enchufe2.java: es la clase que gestiona el control sobre el enchufe 2 y da acceso a los ajustes de la aplicación.
- Enchufe3.java: es la clase que gestiona el control sobre el enchufe 3 y da acceso a los ajustes de la aplicación.
- Enchufe4.java: es la clase que gestiona el control sobre el enchufe 4 y da acceso a los ajustes de la aplicación.
- AccesoAjustes.java: contiene el código del apartado que da acceso a los ajustes para los usuarios supervisores que conocen el código de seguridad.
- AjustesActivity.java: es la clase que contiene la programación que da acceso a cada una de las opciones que personalizan y configuran todos los enchufes.
- SeleccionActivity.java: es la clase encargada de recibir información del usuario sobre qué enchufe quiere actuar, ya sea para personalizarlo, restablecerlo o programar un evento sobre él.
- NombreEnchufe.java: contiene el código que permite al usuario personalizar un enchufe, introduciendo el nombre del aparato a conectar y una imagen o pictograma desde la galería o la cámara.
- NotificacionActivity.java: es la clase que permite al usuario gestionar eventos de notificaciones y apagado de enchufes y establecerlos en momentos determinados.
- AlarmReceiver.java: es la clase que contiene el servicio que actúa en segundo plano permitiendo que los eventos sean lanzados, aunque la aplicación no esté en ejecución.

4.2.3 Implementación en aplicación *Quiero Decirte*

La aportación principal de este proyecto es incluir las funcionalidades de la regleta inteligente en la aplicación *Quiero Decirte*, un comunicador Android que mediante una serie de pictogramas, imágenes y locuciones permite la comunicación de personas con nivel cognitivo bajo, como personas con TEA, mejorando así sus relaciones personales y sus capacidades comunicativas y proporcionándoles un mayor grado de autonomía.

La integración se ha realizado de manera que un usuario supervisor podrá habilitar o deshabilitar el acceso a la aplicación *Control Remoto* desde la pantalla de ajustes del comunicador. Si el usuario supervisor ha habilitado la opción de acceso a la aplicación, en su pantalla principal, donde se encuentran las categorías de los pictogramas, aparecerá un nuevo pictograma con una regleta de enchufes que al ser presionada mostrará inmediatamente la versión integrada de la aplicación de gestión de enchufes siguiendo el mismo esquema presentado en apartados anteriores, manteniendo la aplicación comunicadora todas sus funcionalidades y pantallas intactas.

Si el usuario supervisor no habilita la opción de acceso a la aplicación de gestión de enchufes el comunicador se mantendrá exactamente igual a su última versión

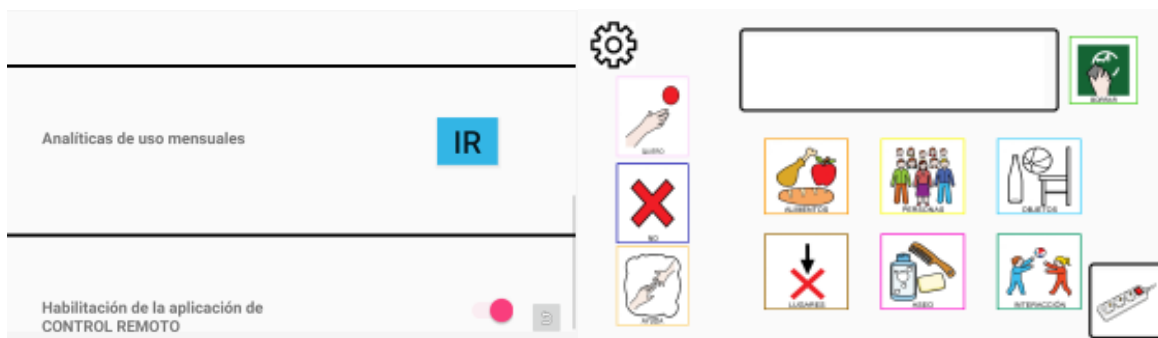


Figura 4-19: Implementación de la aplicación en el comunicador Quiero Decirte

4.3 Desarrollo de la regleta de enchufes inteligentes

La regleta de enchufes inteligentes es el sistema encargado de recibir información de la aplicación Android, procesarla y actuar sobre cada uno de los enchufes mediante un conjunto de elementos electrónicos.

A continuación, pasaré a comentar todos los elementos que han sido necesarios y su funcionalidad para crear esta regleta de enchufes inteligentes.

4.3.1 Elementos de la regleta de enchufes inteligentes

4.3.1.1 Raspberry Pi 3 Modelo b+

La Raspberry Pi 3 Modelo b+ es una placa de ordenador de un tamaño reducido con sistema operativo propio familia de las Raspberry Pi. En ella se pueden cargar varios sistemas operativos y desarrollar una gran cantidad de funcionalidades distintas, como programas complejos, desarrollar juegos o acceder a Internet.

Es necesario cargar e instalar el sistema de instalación de sistemas operativos NOOBS [19] (New Out Of Box Software) mediante una tarjeta microSD en la ranura correspondiente de la Raspberry Pi, configurando e instalando el software necesario para que la placa esté capacitada para desarrollar todas las funcionalidades.

Las características fueron desarrolladas en el apartado 3.3, ideales para desarrollar las funcionalidades del proyecto de una manera rápida y segura.

Esta placa de ordenador será la encargada de recibir y enviar datos bilateralmente con los dispositivos Android que ejecuten la aplicación y actuar sobre los enchufes.

4.3.1.2 Elementos de interacción con la Raspberry Pi Modelo b+

La programación de la Raspberry Pi Modelo b+ se realiza de forma idéntica que con un ordenador personal. La interacción con la placa se efectúa mediante un cable HDMI conectado a una pantalla para visualizar la interfaz de su sistema operativo y con un teclado y ratón conectados a los puertos USB para acceder a sus programas, poder escribir líneas de código y configurar sus características.

4.3.1.3 Protoboard, cables, leds y resistencias

Cuando el usuario encienda o apague uno de los enchufes se encenderá o apagará un led correspondientemente, cada uno de un color para diferenciarlos y poder así personalizarlos de una manera más sencilla desde la aplicación.

Para ello será necesario conectar en una protoboard los leds a los pines GPIO de la Raspberry Pi mediante unos cables del tipo hermbra-macho. La corriente positiva de los leds es de entre 5mA y 20 mA, y la salida de los pines GPIO es de 3,3 voltios, con lo cual las resistencias que tendrán que ir conectadas a la alimentación de los GPIO de la placa serán de 100 Ω para ofrecer mayor seguridad al circuito.

4.3.1.4 Módulo de relé de 4 canales

El módulo de relé es el dispositivo que funcionará de interfaz entre la Raspberry Pi y los enchufes de la regleta inteligente. Como los enchufes soportan un voltaje AC de hasta 250 voltios y una corriente de 10 amperios y la placa solo puede otorgar un voltaje de 3,3 voltios y una corriente de 3 mA, será necesario que los enchufes estén alimentados con la corriente que ofrecen los enchufes de los domicilios y edificios, pero controlados con los GPIO de la Raspberry Pi a través del módulo de relés, que funcionarán como un interruptor cortando o abriendo la corriente de los enchufes según sean alimentados o no por la placa. El módulo de Relés cuenta con 4 canales, uno para cada enchufe, con sus 4 entradas correspondientes conectadas a los GPIO de la Raspberry Pi mediante cables hembra-hembra, una alimentación de 5 voltios alimentada por la placa y un pin de tierra.



Figura 4-20: Modulo de relé de 4 canales

4.3.1.5 Bases de enchufes modulares

Para la fabricación de la regleta inteligente serán necesarios una serie de enchufes modulares con tres entradas para que sean completamente compatibles con los cables de instalación eléctrica y tengan los tres canales: el amarillo y verde de tierra, el marrón de fase y el azul neutro.

Estos enchufes serán controlados por la aplicación Android e irán conectados a la instalación eléctrica y al módulo de relés.



Figura 4-21: Base de enchufe modular

4.3.1.6 Cable de instalación eléctrica

El cable de instalación eléctrica alimentará a los enchufes de la regleta inteligente. Contará con el cable de tierra, el de fase y el neutro. El cable de fase será el que pase por las entradas y salidas del módulo de relés para que este dispositivo controle si deja pasar la corriente para que el enchufe esté apagado o encendido.

4.3.2 Diseño del sistema

Una vez que se dispone de todos los elementos que formarán parte de la regleta de enchufes inteligentes es necesario diseñar el circuito que hará que el sistema funcione correctamente.

El elemento principal es la Raspberry Pi 3 Modelo b+, ya que será la que interactuará con los usuarios a través de la aplicación de Android mediante la red Wifi e intervendrá sobre los enchufes de la regleta. Mediante sus pines GPIO [20] actuará sobre los diferentes circuitos de la placa protoboard para encender o apagar los leds identificativos de los enchufes correspondientes y también sobre el módulo de relés que dejarán pasar la corriente hacia los enchufes habilitándolos o deshabilitándolos.

Para optimizar recursos, se utilizará en el sistema cinco enchufes modulares, cuatro de ellos disponibles para el uso de los usuarios y su control y otro para conectar a la alimentación la Raspberry Pi, ya que necesita 5 voltios y 2,5 amperios, siendo visibles al usuario solamente los cuatro disponibles de la regleta.

La Raspberry Pi también actuará sobre el módulo de 4 relés, que harán de puente en los cables de fase de la alimentación de los enchufes otorgándoles corriente cuando el usuario así lo desee.

El esquemático del diseño del sistema es el siguiente:

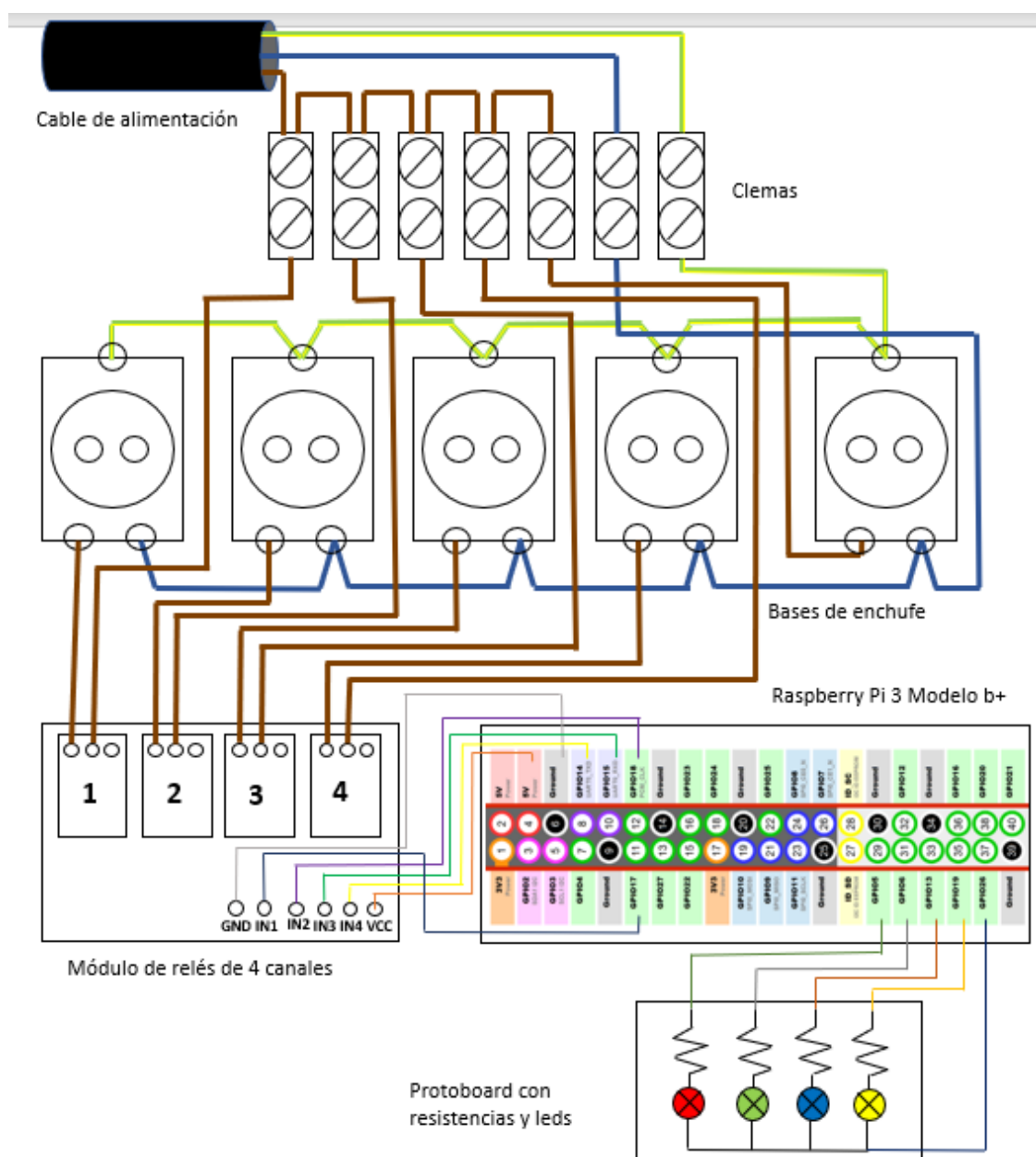


Figura 4-22: Esquemático del sistema de la regleta de enchufes inteligentes

4.4 Desarrollo de la conexión entre la aplicación Android y la regleta de enchufes inteligentes

Para establecer la conexión entre la Raspberry Pi y la aplicación Android es esencial que la placa de ordenador reducido esté conectada a la misma red Wifi que los dispositivos con los que el usuario controlará la regleta de enchufes inteligentes. Por ello será necesaria una configuración inicial en la que se acceda a la Raspberry Pi mediante un ratón, teclado y pantalla para conectarse a la red e introducir la contraseña del router, obteniendo así desde su propia interfaz su dirección IP a la que se enviarán los datos.

En la propia Raspberry Pi se ha desarrollado un fichero de Python, *server.py*, que hace la función de servidor, mediante el entorno de desarrollo *Thonny Python IDE* y que ha sido configurado para que se ejecute automáticamente modificando el fichero de arranque local de la placa */etc/rc.local* mediante el comando *setsid*, que ejecuta un programa en una nueva sesión de la siguiente manera:

```
sleep 60
setsid /usr/bin/python3 /home/Desktop/server.py
```

Es necesario esperar 60 segundos antes de ejecutar el programa del servidor para que la Raspberry Pi tenga el suficiente tiempo para establecer la conexión Wifi con el router.

El protocolo de red que se ha establecido para establecer la conexión entre los dispositivos es el TCP, ya que ofrece un control sobre los datos que se envían detectando los paquetes descartados por la red y reenviándolos y entrega todos sus datos en orden, a diferencia del protocolo UDP. A continuación, pasaré a comentar las partes esenciales del código del servidor que permiten la conexión con los dispositivos Android desde la parte de la Raspberry Pi.

```
1  import socket
2  import sys
3
4  HOST=''
5  PORT = 21570
6
7  # Creación del socket
8  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9  print('Socket Creado')
10 print('.....')
11
12 # Enlazado del socket al puerto
13 sock.bind((HOST, PORT))
14 # Habilitación de escucha
15 sock.listen(5)
16
17 while True:
18     # Espera a aceptar la conexión
19     conn, addr = sock.accept()
20     print('Conectado con ' + addr[0] + ':' + str(addr[1]))
21
22     while True:
23         # Recibe los datos del cliente
24         data = conn.recv(1024)
25         data_decod = data.decode()
26         #print(data_decod)
27
28         # Establece la conexión
29         if(data_decod == '0'):
30             print('Conexión establecida con Éxito')
31             reply = 'ack'.encode()
32             conn.sendall(reply)
33             break
34
35     conn.close()
36 sock.close()
```

Figura 4-23: Código Python del servidor en la Raspberry Pi

Las dos primeras líneas importan los módulos con las funciones necesarias para desarrollar todas las finalidades de la aplicación. Las dos siguientes establecen los datos del HOST y el puerto que recibirá los paquetes mediante TCP de los dispositivos Android, siendo en este caso el puerto 21570 que no es utilizado por ninguna otra aplicación y queda libre para este proyecto.

En la línea 8 de este código se crea el socket por el que se establecerá la conexión, siendo este el objeto que constituye el mecanismo para la entrega y recepción de paquetes a través de Internet definiéndose por direcciones IP, el protocolo de transporte y las direcciones de

los puertos. El parámetro `socket.AF_INET` establece que el socket funcionará mediante la familia de direcciones de Internet para IPv4 y el parámetro `socket.SOCK_STREAM` define que el protocolo que funcionará sobre este canal será el TCP.

La línea de código 13 asocia el socket con la interfaz de red correspondiente al puerto 21570 mediante la función *bind*, mientras que en la línea 15 permite que el canal creado acepte un total de 5 conexiones, suficientes para que varios dispositivos puedan controlar la regleta de enchufes inteligentes de manera simultánea.

Una vez que el canal donde se enviarán los paquetes bidireccionalmente ha sido creado, se programa un bucle infinito donde la Raspberry Pi estará continuamente esperando a recibir una conexión por parte de los dispositivos de los usuarios y almacenar la dirección del cliente (dirección IP y puerto) y la conexión para poder después enviar respuestas desde el servidor mediante la función *accept* de la línea 19.

Cuando la conexión ha sido aceptada y establecida se desarrolla un bucle para recibir los paquetes de datos que envía el cliente (líneas 22 a 33) y analizar sus peticiones. Mediante la función *recv* se leen los datos recibidos por los clientes en formato de bytes, con un tamaño máximo de 1024 bytes, suficientes para la funcionalidad del proyecto, que serán necesario decodificar para conocer exactamente el mensaje del dispositivo Android mediante la función *decode*.

Finalmente, una vez que se decodifica el mensaje enviado por el cliente el servidor lo analiza para diferenciar si la orden dada por el usuario era establecer la conexión, apagar un enchufe o encenderlo. Si el paquete ha sido analizado y decodificado con éxito el servidor envía una respuesta codificada al cliente mediante la función *sendall* para que la aplicación Android se asegure de que no ha habido ningún fallo con la conexión y la orden enviada a la regleta de enchufes inteligente ha sido realizada correctamente.

Cuando el servidor se detiene, ya sea por fallos o por intención del usuario, la conexión y el socket se cierran para que el puerto 21570 no se quede en escucha y se mantenga ocupado en momentos futuros mediante la función *close* de las líneas 35 y 36.

Una vez que el servidor es capaz de abrir un socket TCP, recibir paquetes y enviarlos, se programa la conexión en la parte del cliente, es decir, en la aplicación Android que utilizarán los usuarios. Las partes esenciales de la conexión en la parte del cliente son las siguientes:

```

303 // Función asíncrona que establecerá la conexión
304 public class Socket_AsyncTask extends AsyncTask<Void, Void, Void> {
305     Socket socket;
306     boolean flag = false;
307
308     @Override
309     protected Void doInBackground(Void... params) {
310         try {
311             // Obtención de La dirección IP de La Raspberry Pi
312             InetAddress inetAddress = InetAddress.getByName(wifiModuleIp);
313             // Creación del Socket asociándolo a la dirección IP y al puerto de La Raspberry Pi
314             socket = new java.net.Socket(inetAddress, wifiModulePort);
315             // Generación de Los datos de salida que serán enviados al servidor
316             DataOutputStream dataOutputStream = new DataOutputStream(socket.getOutputStream());
317             dataOutputStream.writeBytes(CMD);
318
319             // Lectura de Los paquetes de entrada recibidos por el servidor
320             DataInputStream dis2 = new DataInputStream(socket.getInputStream());
321             InputStreamReader disR2 = new InputStreamReader(dis2);
322             BufferedReader br = new BufferedReader(disR2);
323
324             // Si el Buffer Leído es un ack significa que La conexión se ha establecido
325             if(br.readLine().equals("ack")){
326                 SharedPreferences prefs = getSharedPreferences(SharedPrefsName, mode: 0);
327                 SharedPreferences.Editor editor = prefs.edit();
328                 editor.putInt("conexion", 1);
329                 editor.commit();
330                 flag = true;
331                 dis2.close();
332                 socket.close();
333             }else{
334                 flag = false;
335                 dis2.close();
336                 socket.close();
337             }
338         } catch (UnknownHostException e) {
339             e.printStackTrace();
340         } catch (IOException e) {
341             e.printStackTrace();
342         }
343         return null;
344     }

```

Figura 4-24: Código de la aplicación Android en Java que establece la conexión con la Raspberry Pi

Desde el software de desarrollo Android Studio se genera una función asíncrona cada vez que el dispositivo quiera mandar una orden a la placa controladora de la regleta permitiendo así enviar datos a la Raspberry Pi en segundo plano sin alterar la ejecución.

En las primeras líneas, desde la 311 a la 314, se genera el socket de la misma manera que en el servidor de la Raspberry Pi, asociándolo a la dirección IP de la placa y a su puerto, el 21570, para establecer el canal de la conexión mediante el protocolo TCP y enlazarlo.

En las líneas 316 y 317 se generan los paquetes de salida que serán enviados a la Raspberry Pi a través del socket generando un buffer de salida asociado y escribiendo en él un texto codificado en bytes enviándolo directamente. Dependiendo del valor del parámetro CMD, se enviarán unos códigos que el servidor decodificará para realizar la funcionalidad que el usuario haya seleccionado.

En las líneas 320 a 322 el cliente crea un buffer de entrada asociado al socket donde se quedará escuchando en el canal hasta recibir la respuesta del servidor, si lee los datos decodificados y la respuesta ha sido un *ack* significará que la Raspberry Pi ha recibido correctamente la orden y la conexión ha sido exitosa, y, en caso contrario, si no recibe la confirmación el cliente entenderá que ha ocurrido un error y pedirá al usuario que compruebe si la dirección IP introducida en la conexión es la correcta.

Una vez que la conexión se ha establecido con éxito y ha habido un intercambio de paquetes entre cliente y servidor, la conexión y el socket se cierran y la funcionalidad principal del proyecto, que es controlar una regleta de enchufes inteligentes mediante una aplicación Android para teléfonos, tabletas y relojes inteligentes, habrá sido realizada satisfactoriamente y el sistema funcionará.

5 Integración, pruebas y resultados

5.1 Integración

Una vez que la aplicación ha sido desarrollada para teléfonos, tabletas y relojes inteligentes Android, la Raspberry Pi 3 Modelo b+ ha sido programada como servidor y el circuito de la regleta de enchufes inteligentes ha sido fabricada exitosamente, se procede a integrar todo el sistema como un único ecosistema, a su utilización como prototipo y a su integración en el comunicador Android *Quiero Decirte*. Para ello ha sido necesario realizar los siguientes pasos.

5.1.1 Integración de la aplicación Android

El objetivo principal del proyecto es implementar una nueva funcionalidad al comunicador *Quiero Decirte* que facilite la realización de tareas de la vida cotidiana a personas con un nivel cognitivo bajo, para ello, tal y como se ha explicado en la sección 4.2.3, se ha integrado la aplicación de gestión y control de una regleta de enchufes inteligente a dicho comunicador Android con éxito. Los usuarios, aparte de comunicarse mediante pictogramas y locuciones con otras personas, también podrán comunicarse con una regleta de enchufes inteligente gestionadas por un supervisor de una manera cómoda y accesible.

Para que la aplicación soporte el registro de usuarios y su inicio de sesión para acceder a los ajustes sobre los enchufes y su conexión es necesario integrarla en una base de datos externa que en el caso de este proyecto será la plataforma de desarrollo de aplicaciones de Google Firebase [21].

Para ello, es necesario registrarse en la consola de Firebase y crear un nuevo proyecto incluyendo información de la aplicación como el nombre del proyecto y su paquete identificador generado por el desarrollador. Una vez que la aplicación ha sido registrada, Firebase generará un fichero JSON que deberá ser incluido entre los ficheros del proyecto junto con las librerías que proporciona esta herramienta. Finalmente, cuando la herramienta ha sido integrada completamente a la aplicación, utilizando las funciones que su librería proporcionan se puede interactuar con su base de datos a tiempo real añadiendo registros y consultándolos desde cualquier dispositivo. Esta herramienta proporciona una gran comodidad a la aplicación ofreciendo al usuario seguridad sobre cambios no deseados de la regleta.

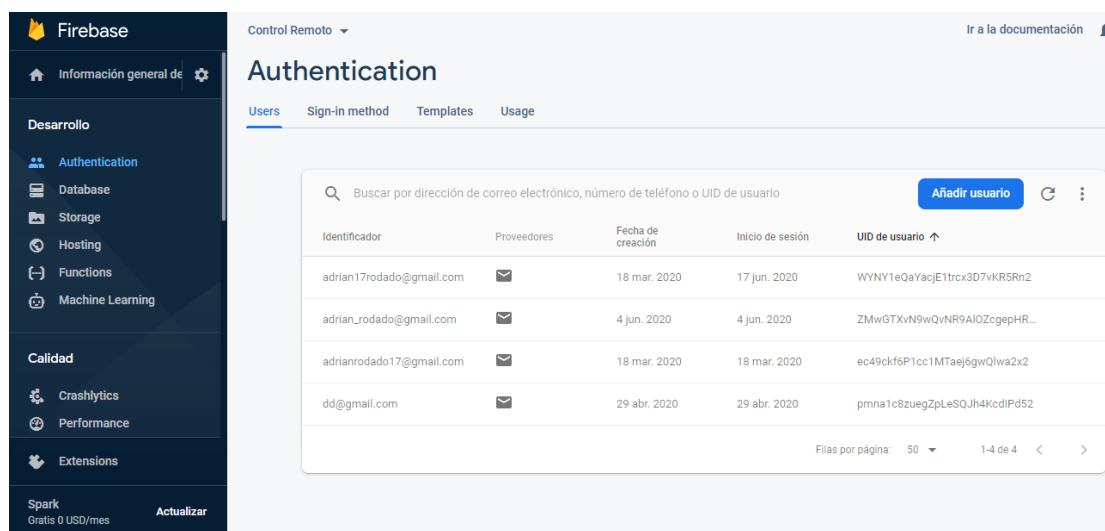


Figura 5-1: Base de datos de usuarios de la herramienta Firebase

Cuando la aplicación se ha desarrollado con éxito y es capaz de ser soportada por toda la familia de dispositivos Android para los que va dirigida este proyecto es necesario generar el archivo APK para poder realizar e instalar la aplicación en diferentes dispositivos sin necesidad de subir la aplicación a la Google Play Store.

El archivo APK es el archivo ejecutable de las aplicaciones para Android que empaqueta todo su código utilizado para distribuir e instalar la aplicación en los dispositivos que soporten sus requisitos, es un fichero que los desarrolladores firman digitalmente garantizando su autoría y puede ser transferido entre dispositivos o directamente publicado en la Google Play Store. En caso de este proyecto se ha generado un archivo APK a través del propio software de Android Studio desde su opción correspondiente para instalarlo directamente en varios dispositivos para realizar las pruebas en vez de publicarlo en la Google Play Store, ya que sin un prototipo de la regleta no tendría sentido su uso.

5.1.2 Integración del circuito de la regleta de enchufes inteligentes

Finalmente, una vez que se ha desarrollado el esquema del circuito y estableciendo las salidas de los GPIO de la Raspberry Pi tal y como están diseñados en el esquema del apartado 4.3.2 y han sido todas ellas programadas correctamente se pasa a fabricar el circuito final estableciendo todas las conexiones entre todos los elementos necesarios. El circuito de la regleta de enchufes inteligentes final es el siguiente:

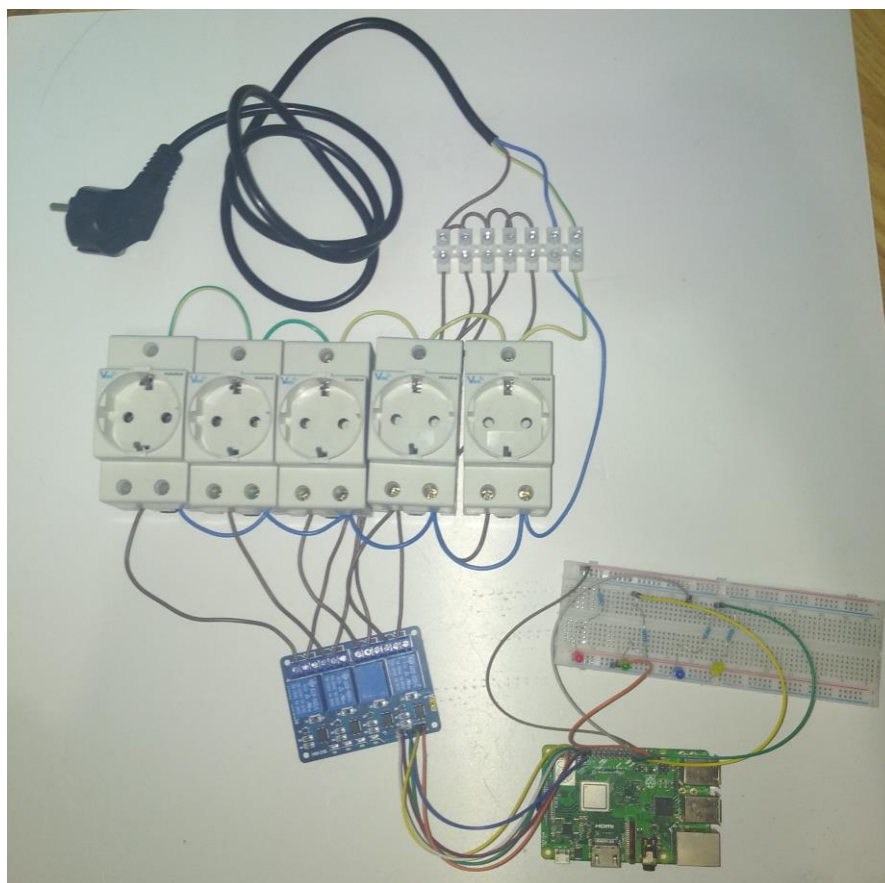


Figura 5-2: Circuito completo de la regleta de enchufes inteligentes

5.2 Pruebas

Una vez que todo el proyecto ha sido desarrollado satisfactoriamente se instala el APK de la aplicación en diferentes tabletas y teléfonos y se conecta la Raspberry Pi a la red Wifi de mi domicilio. Es necesario que el dispositivo Android y el servidor estén conectados a la misma red para que puedan abrir el socket y enviar y recibir paquetes. En el código del servidor de la placa se implementan una serie de líneas que irán mostrando información por terminal para ver qué ocurre durante la ejecución y si la conexión y los datos se establecen de manera correcta.

La campaña de pruebas más importante será realizada por el equipo de expertos de la Unidad de Educación Especial del colegio Romero Peña de La Solana, en Ciudad Real, y por un grupo de alumnos con un nivel cognitivo bajo.

La salida que ofrece el servidor por terminal, la actuación sobre los enchufes y la recepción de la información desde los dispositivos Android se comentarán en el siguiente apartado.

5.3 Resultados

Antes de comentar los resultados obtenidos mediante la integración del sistema completo comentaré una serie de resultados que ofrece la herramienta de Firebase integrada en la aplicación Android, ya que ofrece una serie de estadísticas y datos de su utilización aparte de implementar una base de datos de usuarios.

Firestore ofrece información útil, que se puede utilizar de realimentación sobre la aplicación para desarrollar futuras versiones, como el número de usuarios activos de la aplicación en determinados rangos de tiempo, cómo se han adaptado los usuarios a cada una de las versiones de la aplicación, de dónde son los usuarios, cuántos eventos realizan y cuánto tiempo pasan en cada pantalla.

En este caso, donde el único usuario actualmente es el programador por disponer del prototipo de la regleta de enchufes inteligentes, uno de los datos más relevantes es el tiempo de permanencia en cada pantalla. Mediante la información ofrecida por Firestore se puede observar que la pantalla más utilizada es la que permite al usuario controlar los enchufes de la regleta ya que es la finalidad principal de la aplicación.



The screenshot shows the 'Interacción de los usuarios' (User Interaction) section in Firebase Analytics, specifically the 'Clase de pantalla' (Screen Class) report. The table lists various screen classes with their percentage of total usage, change from the previous period, and average time spent on each screen.

Clase de pantalla	% del total		Tiempo medio	
HomeActivity	69,75 %	↑ 1... %	0 mi...1 s	↑ 500,0 %
LoginActivity	15,59 %	↑ 32... %	0 mi...3 s	↑ 164,1 %
AjustesActivity	8,77 %	↓ 88,6 %	0 mi...0 s	↓ 87,5 %
AddEnchufeActivity	3,53 %	↓ 61,2 %	0 mi...1 s	↓ 46,3 %
NotificationActivity	2,16 %	↑ 28,0 %	0 mi...5 s	↑ 77,0 %
MainActivity	0,2 %	↓ 83,4 %	0 mi...0 s	↓ 89,5 %
AnaliticasActivity	0 %	↓ 10... %	0 mi...0 s	↓ 100,0 %

Figura 5-3: Datos de interacción de los usuarios de la aplicación en la herramienta de Firebase

Respecto a los resultados obtenidos mediante la realización de pruebas del ecosistema completo de la aplicación Android con la regleta de enchufes inteligentes, han sido obtenidos visualizando la salida de la terminal de la Raspberry Pi cuando se interactúa con ella mediante la red Wifi y obtiene paquetes de datos a través del socket, mediante la información obtenida en la interfaz de usuario de la aplicación Android en distintos tipos de dispositivos al recibir paquetes de confirmación de la placa y observando si el enchufe se activa y desactiva cuando se da la orden desde el dispositivo Android.

Al tratar de establecer la conexión entre la aplicación Android y la Raspberry Pi introduciendo su dirección IP desde la pantalla de ajustes, la salida por terminal de la placa y la interfaz de usuario del dispositivo Android es la siguiente, donde se puede ver que la Raspberry Pi recibe datos desde el dispositivo y la aplicación recibe la confirmación desde la placa:



Figura 5-4: Resultados en la consola de la Raspberry Pi y en la interfaz de la aplicación al establecer la conexión

Al intentar encender y apagar cada uno de los enchufes mediante la aplicación Android, una vez que la conexión se ha establecido, la terminal de la Raspberry Pi muestra los siguientes resultados y los enchufes reaccionan de la siguiente manera:

```
pi@raspberrypi:~/Desktop $ python3 server.py
Socket Creado
.....

Conectado con 192.168.1.38:46396
Conexion Establecida con Éxito

Conectado con 192.168.1.38:46398
Enchufe 1 activado

Conectado con 192.168.1.38:46400
Enchufe 1 desactivado

Conectado con 192.168.1.38:46402
Enchufe 2 activado

Conectado con 192.168.1.38:46404
Enchufe 2 desactivado

Conectado con 192.168.1.38:46406
Enchufe 3 activado

Conectado con 192.168.1.38:46408
Enchufe 3 desactivado

Conectado con 192.168.1.38:46410
Enchufe 4 activado

Conectado con 192.168.1.38:46412
Enchufe 4 desactivado
```

Figura 5-5: Resultados de la consola de la Raspberry Pi al recibir datos de la aplicación Android

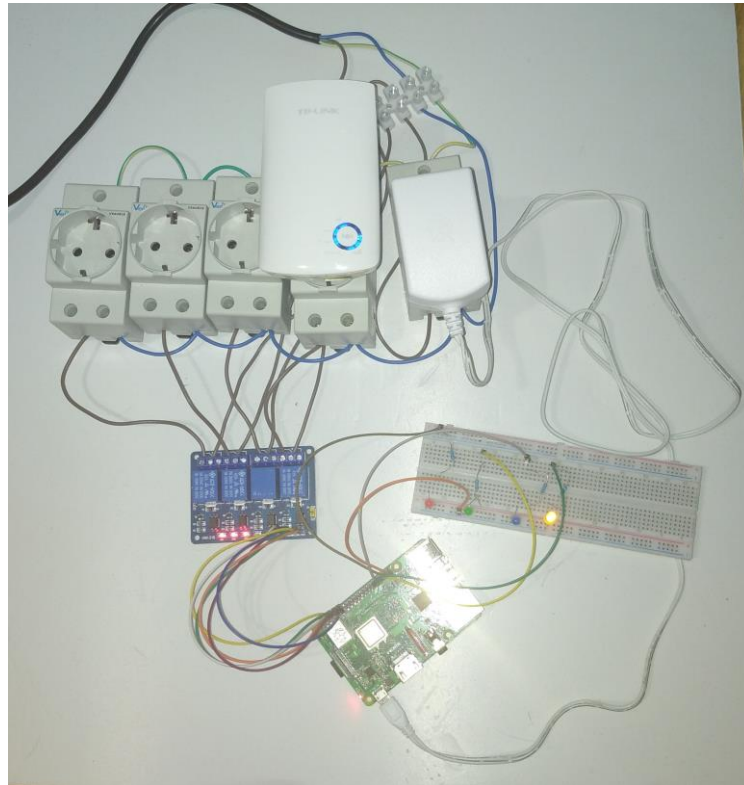


Figura 5-6: Resultado de la interacción entre la aplicación Android y la Raspberry Pi al intentar encender uno de los enchufes de la regleta inteligente de manera remota

Pudiendo comprobar que la Raspberry Pi recibe todas las ordenes correctamente desde el comunicador Android, ya sean de establecimiento de conexión, de encender cualquiera de los enchufes o apagarlos, y es capaz de actuar según la orden recibida.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Este proyecto consiste en integrar en el comunicador Android *Quiero Decirte*, aplicación destinada a mejorar y potenciar la comunicación y las relaciones personales de personas con un nivel cognitivo bajo, tales como personas con TEA u otros tipos de discapacidades que dificultan la comunicación efectiva y las relaciones personales, una nueva funcionalidad para así ofrecerles mayor autonomía y comodidad para realizar tareas de la vida cotidiana, tales como encender cualquier dispositivo eléctrico.

Para ello, se ha desarrollado un ecosistema de dispositivos Android como teléfonos, tabletas y relojes inteligentes que mediante una aplicación con telemandos sea capaz de gestionar y controlar una regleta de enchufes inteligentes a través de una Raspberry Pi con una interfaz de usuario sencilla, cómoda, intuitiva y principalmente accesible para que pueda ser utilizada por cualquier persona independientemente de su nivel cognitivo e incluirla en dicho comunicador.

Para llevar a cabo el objetivo de este Trabajo de Fin de Máster se ha desarrollado un proyecto con las siguientes características:

- Diseño y desarrollo de una aplicación Android disponible para teléfonos, tabletas y relojes inteligentes capaz de gestionar y controlar una regleta de enchufes inteligentes mediante una red Wifi.
- Diseño de una interfaz de usuario de la aplicación sencilla, con pocos estímulos, intuitiva y accesible.
- Un alto nivel de personalización de la aplicación y de la regleta de enchufes mediante nombres, imágenes y pictogramas.
- Acceso a una base de datos desde la aplicación Android de registro de usuarios.
- Acceso a analíticas sobre el uso de cada uno de los enchufes.
- Implementación e integración de la aplicación Android en un comunicador, *Quiero Decirte*, destinado a ayudar a la comunicación e interacción de personas con discapacidades que impliquen un nivel cognitivo bajo, como el trastorno del espectro autista o deficiencia intelectual.
- Desarrollo de un servidor en una Raspberry Pi capaz de recibir y enviar paquetes de datos a través de una conexión Wifi mediante el protocolo TCP con una velocidad de transmisión alta y seguridad mediante protección de pérdida de paquetes.
- Diseño e implementación de un circuito que permite a la Raspberry Pi actuar sobre cada uno de los enchufes de manera individual y conjunto a través de un módulo de relés.

En cuanto a aspectos formativos, los principales conocimientos adquiridos mediante la realización de este trabajo de fin de máster son los siguientes:

- Adquisición de conocimientos sobre el lenguaje de programación Java.
- Adquisición de conocimientos sobre el lenguaje de programación XML.
- Lenguaje de programación Python.
- Instalación, utilización y programación de una Raspberry Pi.

- Adquisición de conocimientos sobre el entorno Android Studio.
- Utilización del entorno Thonny Python IDE.
- Implementación de una base de datos de Firebase en una aplicación Android.
- Control de relés y actuación sobre elementos electrónicos.
- Envío de paquetes a través de una conexión Wifi real mediante el protocolo TCP.

Este proyecto ha sido realizado durante siete meses con la intención de continuar lanzando versiones mejorando la aplicación según las necesidades de los usuarios y añadiendo nuevas funcionalidades al comunicador *Quiero Decirte*. Respecto al código realizado en el entorno de Android Studio, ha sido necesario programar un total de 22 ficheros Java con 4000 líneas de código, 30 ficheros XML con 3200 líneas de código y un fichero Python con 110 líneas de código.

6.2 Trabajo futuro

Como trabajo futuro se podrían realizar una serie de mejoras al sistema añadiendo nuevas características y funcionalidades a la aplicación Android y a la regleta de enchufes inteligentes y poder llegar a un gran número de usuarios:

- Traducción de la aplicación a nuevos idiomas como el inglés, alemán, francés e italiano.
- Programación de escenarios para que el usuario pueda apagar o encender enchufes según características del entorno como temperatura o humedad.
- Realización de una versión de la aplicación disponible para el sistema operativo iOS.
- Añadir puertos USB a la regleta de enchufes inteligentes.
- Reducir el tamaño de las bases de enchufe para reducir así el tamaño de la regleta.
- Realización de una versión compatible con asistentes de voz como Alexa o Google Home.

Referencias

- [1] Adrián Rodado Peribáñez, “Comunicador Android y Herramienta de Análisis de Uso”, Escuela Politécnica Superior, Junio 2018, <https://repositorio.uam.es/handle/10486/688441>
- [2] Distribuidor de la regleta inteligente Wifi LOONFREE, LOONFREE, Amazon, 7 de Enero de 2018, <https://www.amazon.es/inteligente-enchufes-Interruptor-temporizador-Smartphone/dp/B078NTMXZ9>
- [3] Distribuidor de la regleta inteligente Wifi TAOCOCO, TAOCOCO, Amazon, 1 de Agosto de 2019, <https://www.amazon.es/TAOCOCO-inteligente-Temporizador-aplicaci%C3%B3n-Compatible/dp/B07VXW29TJ/>
- [4] Tuya Inc., Google Play Store, última actualización 22 de Junio de 2020, <https://play.google.com/store/apps/details?id=com.tuya.smartlife&hl=es>
- [5] TP-LINK Research America, Google Play Store, última actualización 9 de Junio de 2020, https://play.google.com/store/apps/details?id=com.tplink.kasa_android&hl=es
- [6] Plataforma StatCounter, StatCounter GlobalStats, Junio de 2020, <https://gs.statcounter.com/os-market-share/mobile-tablet/worldwide/#quarterly-201903-202002>
- [7] Android Developers, “Panel de Distribución”, última actualización 15 de Junio de 2020, <https://developer.android.com/about/dashboards>
- [8] The NetCircle Network, “An Introduction to Android Basics, Screen Sizes”, 26 de Agosto de 2015, <https://thenetcircle.com/updates/introduction-android-basics/>
- [9] Mario Pablo Mata Ramírez, “Tecnología Bluetooth”, Monografías, <https://www.monografias.com/trabajos43/tecnologia-bluetooth/tecnologia-bluetooth.shtml>
- [10] Isaura Santana, “Red Inalámbrica WIFI”, <https://redwifi.wordpress.com/caracteristicas/>
- [11] Portal de Arduino, Arduino, 2020, <https://www.arduino.cc/>
- [12] Arduino MKR WIFI 1010, Arduino, Septiembre de 2018, <https://store.arduino.cc/arduino-mkr-wifi-1010>
- [13] Arduino UNO WIFI Rev 2, Arduino, Diciembre de 2018, <https://store.arduino.cc/arduino-uno-wifi-rev2>
- [14] Portal de Raspberry Pi, Raspberry Pi Foundation, 2020, <https://www.raspberrypi.org/>
- [15] Raspberry Pi 3 Modelo a+, Raspberry Pi Foundation, Marzo de 2018, <https://www.raspberrypi.org/products/raspberry-pi-3-model-a-plus/>
- [16] Raspberry Pi 3 Modelo b+, Raspberry Pi Foundation, Marzo de 2018, <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [17] Raspberry Pi 3 Modelo b, Raspberry Pi Foundation, Marzo de 2016, <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [18] Android Developers, “Servicios”, última actualización 27 de Diciembre de 2020, <https://developer.android.com/guide/components/services>
- [19] New Out Of Box Software (NOOBS), Raspberry Pi, última actualización 14 de Febrero de 2020, <https://www.raspberrypi.org/documentation/installation/noobs.md>
- [20] Documentación GPIO, Raspberry Pi, <https://www.raspberrypi.org/documentation/usage/gpio/>
- [21] Consola de Firebase, Google, <https://firebase.google.com/>

Glosario

API	Application Programming Interface
APK	Android Application Package
TEA	Trastorno del Espectro Autista
TCP	Transmission Control Protocol
ACK	Acknowledgments
JSON	JavaScript Object Notation
DPI	Dots Per Inch
IDE	Integrated Development Environment

Anexos

A Manual de usuario

Al iniciar la aplicación, ya sea desde la versión individual desde un teléfono, una tableta o un reloj inteligente, o desde la integración del comunicador *Quiero Decirte*, aparecerá durante tres segundos una pantalla de presentación de la aplicación donde se muestra el logotipo y el nombre.

El funcionamiento de la aplicación será diferente dependiendo de si el dispositivo que la ejecuta es una tableta y un teléfono o si es un reloj inteligente. En el caso de tabletas y teléfonos el funcionamiento será el siguiente:

La pantalla principal tendrá una lista de los enchufes con sus imágenes y nombres asociados junto a unos *switches* que establecerán el estado de cada enchufe pudiendo encenderlo o apagarlo si la conexión con la regleta ha sido establecida. Desde esa pantalla deslizando el dedo de izquierda a derecha se mostrará un menú lateral donde el usuario podrá encender o apagar todos los enchufes instantáneamente, podrá iniciar sesión o cerrarla si ha sido registrado previamente, acceder a los ajustes de la aplicación o cerrar la aplicación eliminándola del segundo plano del dispositivo.

Para personalizar los enchufes y acceder a los ajustes será necesario iniciar sesión en la base de datos de usuarios de la aplicación, desde la pantalla principal, donde será necesario introducir un correo electrónico válido y una contraseña y se podrá solicitar la verificación del email mediante un correo enviado por la herramienta de Firebase.

Una vez que el usuario se ha registrado y ha iniciado sesión se podrá acceder a la pantalla de ajustes del sistema. En esta sección se deberá establecer la conexión con la Raspberry Pi introduciendo su dirección IP, accesible desde la configuración del router al que se ha conectado, siendo necesario que el dispositivo Android se conecte a la misma red Wifi que la regleta.

Desde los ajustes también se podrán personalizar y restablecer cada uno de los enchufes, asociándoles un nombre y una imagen o pictograma desde sus apartados correspondientes. También se podrá acceder a las analíticas de cada enchufe donde se podrá analizar cuánto tiempo ha estado activo cada uno y restablecer sus datos en cualquier momento.

En los ajustes de la aplicación también se permite la activación de eventos tales como recibir una notificación en el dispositivo de recordatorio sobre un enchufe o directamente apagarlo o encenderlo en un momento determinado por el usuario mediante un contador de días, horas y minutos. Finalmente, también se permite en esta pantalla bloquear los enchufes para que los usuarios sin acceso a la base de datos de la aplicación no puedan encender o apagar ninguno de los enchufes sin permiso.

La versión de relojes inteligentes funciona exactamente igual, salvo que por el tamaño reducido de la pantalla ha sido necesario acomodar las pantallas a su interfaz, dividiendo la pantalla de los enchufes en cuatro, una para cada uno de los enchufes, siendo accesibles deslizando el dedo por la pantalla de izquierda a derecha y viceversa, y también se ha

prescindido de la opción de acceder a las analíticas sobre los enchufes debido a la gran cantidad de información que aporta en una pantalla tan pequeña.

La conexión inicial de la Raspberry Pi a la red Wifi de la vivienda se realiza mediante su propia interfaz de manera manual, posteriormente la conexión se establecerá automáticamente sin necesidad de realizar ningún paso extra.

En la integración en el comunicador *Quiero Decirte* será necesario que la persona supervisora del usuario que haga uso de la aplicación habilite la opción desde la pantalla de ajustes para que sea accesible desde su pantalla principal teniendo un acceso directo y completamente integrado.

B Pantallas de la aplicación

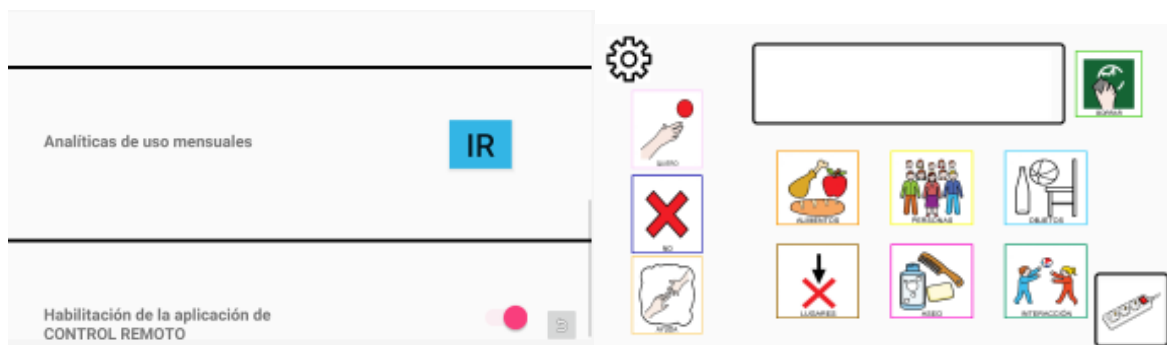


Figura B-1: Pantallas del comunicador QUIERO DECIRTE – Pantalla de ajustes con la opción de habilitación del control remoto – Pantalla principal con el acceso a la sección de control remoto

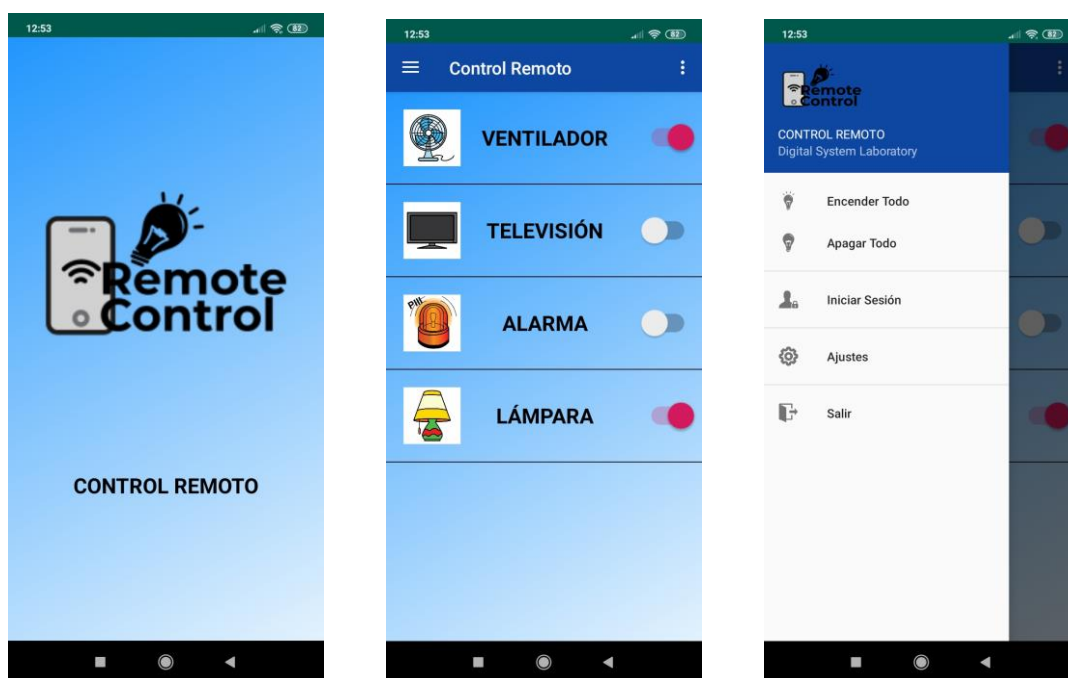


Figura B-2: Pantalla de presentación - Pantalla principal de control de enchufes – Pantalla principal con el menú lateral

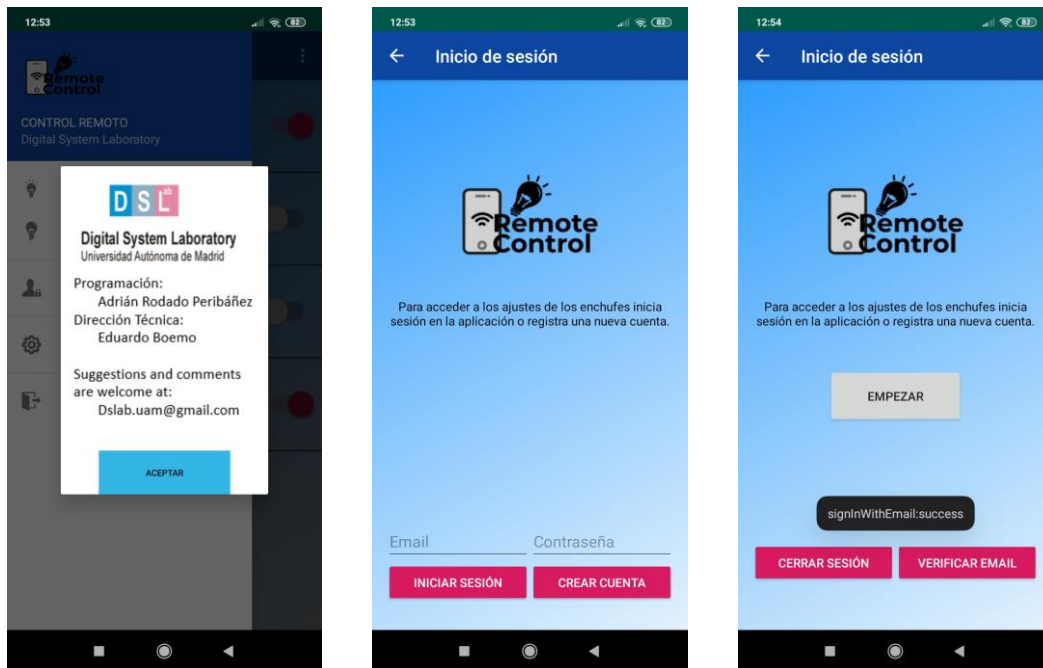


Figura B-3: Fragmento de la información técnica de la aplicación – Pantalla de inicio de sesión o registro – Pantalla de cierre de sesión o verificación de email

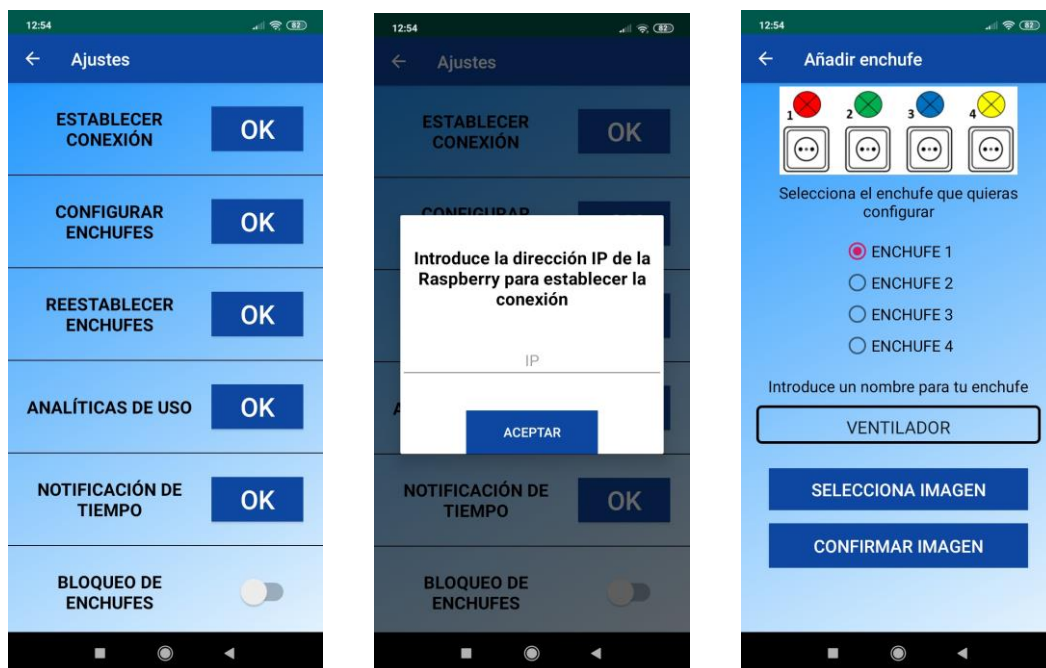


Figura B-4: Pantalla de ajustes del control remoto - Fragmento de establecimiento de la conexión con la Raspberry Pi – Pantalla de personalización de los enchufes de la regleta

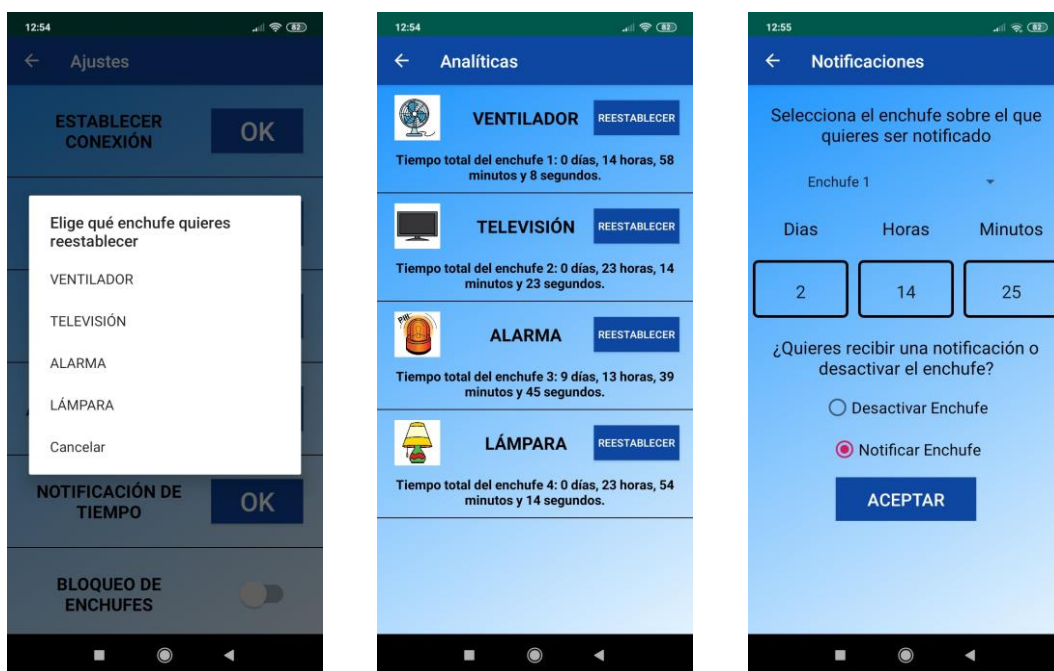


Figura B-5: Fragmento de restablecimiento de la personalización de los enchufes - Pantalla de analíticas de uso sobre cada uno de los enchufes - Pantalla de generación de notificaciones y eventos sobre los enchufes



Figura B-6: Pantalla de presentación en relojes inteligentes – Pantalla de control del enchufe 1 – Pantalla de acceso a los ajustes de la aplicación



Figura B-7: Pantalla de ajustes de la aplicación – Pantalla de conexión con la Raspberry Pi – Pantalla de selección de los enchufes

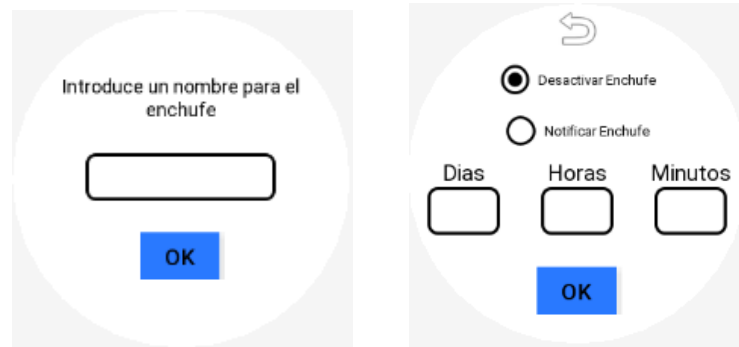


Figura B-8: Pantalla de personalización de los enchufes - Pantalla de generación de notificaciones y eventos sobre los enchufes